

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Filtro de condensadores comutados adaptativo

Miguel António Ferreira de Pina

Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Orientador: Vítor Grade Tavares (Prof. Doutor)

Co-orientador: Pedro Guedes de Oliveira (Prof. Doutor)

Julho de 2009

A Dissertação intitulada

“FILTRO DE CONDENSADORES COMUTADOS ADAPTATIVO”

foi aprovada em provas realizadas em 17/Julho/2009

o júri



Presidente Professor Doutor Mário Jorge Moreira Leitão

Professor Associado do Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto



Professor Doutor Paulo Mateus Mendes

Professor Auxiliar do Departamento de Electrónica Industrial da Escola de Engenharia da Universidade do Minho



Professor Doutor Vitor Manuel Grade Tavares

Professor Auxiliar do Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto



Professor Doutor Pedro Henrique Henriques Guedes de Oliveira

Professor Catedrático do Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projecto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extractos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são correctamente citados.



Autor - MIGUEL ANTÓNIO FERREIRA DE PINA

Faculdade de Engenharia da Universidade do Porto

Resumo

Este documento reporta sobre filtros adaptativos e suas implementações, no domínio analógico com vista à implementação de um filtro Gamma que é o objectivo desta tese. Os filtros adaptativos são utilizados em diferentes aplicações e.g., identificação de sistemas, cancelamento de ecos, etc. Estes filtros conseguem alterar os seus coeficientes de forma a minimizar a diferença entre o sinal obtido e o sinal desejado. Esta variação é possível através da introdução de um algoritmo no filtro, neste trabalho o usado é o LMS. O problema nas implementações dos filtros adaptativos analógicos é a impossibilidade de obter atrasos ideais. Para numa implementação analógica podemos aproximar o atraso ideal, colocando em série vários filtros passa-baixo dando origem ao filtro Gamma.

Nesta tese estão expostos todos os blocos necessários para implementação de um filtro adaptativo utilizando um filtro Gamma com base no algoritmo LMS. Foi implementado um filtro adaptativo em *Matlab/Simulink* para comprovar o seu funcionamento. O esquema e simulação do filtro adaptativo foram realizados em *Cadence*, inicialmente com componentes ideais, passando para a implementação individual de cada bloco e no final a interligação de todos os blocos implementados.

Abstract

This work address adaptative filters and their possible implementations in the analog domain, with a Gamma filter implementation as the main objective of this thesis. Adaptative filters are used in various situations *e. g.*, system identification, echo canceling and many other different applications. This type of filters can change their parameters, so they can minimize the difference between the obtained an the desired signal. This variation of the filter parameters is possible with the help of an algorithm present in the filter. In this work, the LMS algorithm was the selected for implementation. The main issue when implementing analog filters is the impossibility of obtaining ideal delays. An ideal delay can be obtained putting several low-pass filters in series with each other, which is a Gamma filter.

In this work all the necessary circuits to implement an adaptative filter using a Gamma filter with a LMS algorithm are exposed. An adaptative filter was implemented in *Matlab/Simulink* to corroborate the system operation. The electrical circuit and its simulation was performed using *Cadence Design Systems* software. First ideal components were used, and step by step each block of the system was replaced by the final circuit, until all the ideal components were replaced.

Agradecimentos

Inicialmente quero agradecer à minha família pela compreensão dos poucos momentos passados durante este semestre. Gostava de agradecer especialmente à minha mãe e ao meu irmão toda a força demonstrada, já que este foi um ano muito difícil para nós devido ao falecimento do meu pai, a quem dedico todo este trabalho.

Gostava agradecer ao professor Vítor e ao professor Pedro pela ajuda prestada durante o desenvolvimento deste trabalho, tirando dúvidas e sugerindo ideias para a implementação. Sem a vossa ajuda este trabalho tornava-se mais difícil de realizar.

Ao pessoal do núcleo de Micro Electrónica desta faculdade, um obrigado por toda a ajuda e apoio prestada especialmente ao Américo Dias, Cândido Duarte e Daniel Oliveira. Gostava de agradecer a todos os meus amigos pelos bons momentos de descontração passados durante a realização deste trabalho.

Miguel Pina

“No meio da dificuldade encontra-se a oportunidade.”

Albert Einstein

Conteúdo

1	Introdução	1
1.1	Objectivos	2
1.2	Motivação	2
1.3	Estrutura da Dissertação	2
2	Revisão Bibliográfica	5
2.1	Sistemas Adaptativos	5
2.2	Implementações de Filtros Adaptativos	6
2.3	Algoritmos adaptativos	7
2.4	Filtro Gamma	8
2.5	Estrutura de um filtro adaptativo	9
3	Teste de funcionamento e blocos a implementar	11
3.1	Simulação Matlab/Simulink com filtros FIR	11
3.2	Simulação Matlab/Simulink com filtro Gamma	13
3.3	Ferramentas da tecnologia	15
3.4	Filtro Passa-Baixo	15
3.5	Circuito a implementar	18
4	Implementações e Resultados	21
4.1	Cascata de Filtros Passa-Baixo	21
4.2	Filtro multiplexado	23
4.3	Offset	24
4.4	Análise de operacionalidade	24
4.5	Outros blocos necessários	25
4.5.1	Somador - Somatório pesado dos coeficientes	25
4.5.2	Integrador - Actualização dos coeficientes	27
4.5.3	Subtractor - Cálculo do erro	28
4.6	Estrutura final	30
4.7	Simulação em <i>Matlab/Simulink</i> do circuito final ideal	33
4.8	Implementação do Amplificador	35
4.9	Implementação do Switch	37
4.10	Simulações finais	37
4.10.1	Simulação com dois coeficientes	38
4.10.2	Simulação com quatro coeficientes	38
4.11	Compensação dos <i>offsets</i>	40

5	Conclusões e Trabalho Futuro	43
5.1	Conclusão	43
5.2	Trabalho Futuro	44
A	Implementação futura	45
A.1	Variação do μ	45
	Referências	47

Lista de Figuras

2.1	Blocos de um filtro adaptativo	5
2.2	Estrutura de um filtro Gamma	8
2.3	Comportamento do filtro Gamma para diferentes intervalos de μ	9
2.4	Estrutura de um filtro adaptativo	9
3.1	Filtro adaptativo com FIR's	12
3.2	Resultados do filtro adaptativo com FIR's	12
3.3	Coeficientes do filtro adaptativo com FIR's	13
3.4	Filtro adaptativo com filtro Gamma	14
3.5	Resultados do filtro adaptativo com filtro Gamma	14
3.6	Coeficientes do filtro adaptativo com filtro Gamma	15
3.7	Filtro passa-baixo	16
3.8	Sinais de controlo ϕ_1 e ϕ_2	17
3.9	Resposta filtro passa-baixo	18
3.10	Esquema inicial do filtro	19
3.11	Esquema inicial para calculo da saída e coeficientes	20
4.1	Série de três filtros passa-baixo	21
4.2	Cascata de filtros	22
4.3	Sinais de controlo para filtro multiplexado	22
4.4	Três filtros multiplexados	23
4.5	Resposta filtro multiplexado	24
4.6	Resposta filtro multiplexado com <i>offset</i>	25
4.7	Somador duas entradas	26
4.8	Resultados do somador dos atrasos do filtro	26
4.9	Sinais de controlo para somador dos atrasos do filtro	27
4.10	Integrador para actualização dos coeficientes	28
4.11	Resultados do integrador para actualização dos coeficientes	28
4.12	Subtractor para cálculo do erro	29
4.13	Resultados do subtractor para cálculo do erro	29
4.14	Sinais de controlo para subtractor de cálculo do erro	30
4.15	Circuito final ideal	31
4.16	Sinais de controlo para circuito final ideal	32
4.17	Entrada, saída e erro do filtro adaptativo ideal	33
4.18	Coeficientes do filtro adaptativo ideal	34
4.19	Sinais do filtro final em Matlab/Simulink	34
4.20	Coeficientes do filtro final em Matlab/Simulink	35
4.21	Circuito do amplificador	36

4.22	Ganho e fase do amplificador	36
4.23	Circuito do <i>Switch</i>	37
4.24	Sinais do circuito final, dois coeficientes	38
4.25	Coeficientes do circuito final, dois coeficientes	39
4.26	Sinais do circuito final, quatro coeficientes	39
4.27	Coeficientes do circuito final, quatro coeficientes	40
4.28	Compensação de <i>offset</i> no integrador	41
4.29	Compensação de <i>offset</i> no subtrator	41
4.30	Coeficientes do circuito final, quatro coeficientes	42
4.31	Coeficientes do circuito final, quatro coeficientes	42

Lista de Tabelas

2.1	Entrada e saídas de um filtro adaptativo	6
2.2	Complexidade numérica do LMS e do RLS	8
3.1	Sinais de controlo ϕ_1 e ϕ_2	17
4.1	Sinais de controlo para série de Filtros	23
4.2	Sinais de controlo	32
4.3	Tamanhos dos transístores do amplificador	35
4.4	Tamanhos dos transístores do switch	37

Abreviaturas e Símbolos

Abreviaturas

CMOS	Complementary Metal-Oxide Semiconductor
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
LMS	Least Mean Square
RLS	Recursive Least Squares
VLSI	Very Large-Scale Integration

Símbolos

$x(n)$	Sinal de entrada do filtro
$y(n)$	Sinal de saída do filtro
$d(n)$	Sinal desejado
$e(n)$	Sinal de erro
$v(n)$	Sinal de ruído
$\vec{w}_k(n)$	Vector de k coeficientes
μ	Profundidade da memória do filtro Gamma
$\vec{\nabla}(n)$	Gradiente da função erro
∂	Derivada parcial
η	Constante de velocidade do algoritmo

Capítulo 1

Introdução

Actualmente os filtros adaptativos são muito usados em sistemas de telecomunicações, controlo, áudio, radar, entre outros. A filtragem adaptativa é utilizada em múltiplas aplicações como cancelamento de eco, igualizações, identificação de sistemas, cancelamento de ruído e predições. Muitas destas aplicações envolvem processamento de sinal gerados por sistemas cujas características não são conhecidas desde o início do processamento [1] [2].

Um filtro adaptativo é um filtro que ajusta automaticamente a sua função de transferência consoante um algoritmo, aumentando desta forma a performance do sistema. Existem vários algoritmos adaptativos, sendo a maioria deles variantes dos algoritmos LMS e RLS. O algoritmo LMS é usado em filtros adaptativos para encontrar os coeficientes que menor erro quadrático produzem. O algoritmo RLS é utilizado para encontrar os coeficientes que recursivamente minimizam o erro quadrático.

Por exemplo, para o cancelamento do ruído, recorrendo a filtros FIR, é necessário comparar o sinal de saída do filtro ao sinal que ambicionamos obter. A saída do filtro é obtida pelo somatório pesado dos diversos atrasos que o sinal de entrada sofre. A diferença entre o sinal desejado e o sinal de saída do filtro dá a informação do erro a ser minimizado.

Neste tipo de filtros a estabilidade é um factor muito relevante, logo é necessário um controlo da função de transferência destes durante a adaptação. Como a estabilidade é uma tarefa difícil de controlar, recorre-se frequentemente a filtros do tipo FIR, pois garantem uma solução sempre estável. Quando estes filtros FIR são utilizados em sistemas que requerem uma resposta impulsional longa, é necessário o uso de um elevado número de atrasos para obter uma resolução temporal aceitável. Por outro lado, podemos optar por usar filtros IIR, pois estes possuem uma resposta impulsional infinita, exigindo desta forma um menor número de *taps*¹ a usar no sistema. A adaptação dos pólos num filtro IIR pode dar origem a um sistema instável, o que dificulta o processo de adaptação.

¹ Atraso de um filtro passa-baixo.

A implementação de atrasos ideais num sistema analógico é uma tarefa irrealizável. A melhor implementação para os filtros adaptativos passa pelo uso de uma cadeia de filtros passa-baixo, aproximando assim a um conjunto de atrasos ideais. Esta solução é conhecida por filtro Gamma. Este tipo de filtros oferecem um mecanismo que permite ajustar a relação profundidade/resolução da memória do sistema. Como veremos mais à frente, o controlo da estabilidade é trivial, apresentando uma solução de compromisso de um FIR e um IIR.

1.1 Objectivos

Esta tese tem como objectivo implementar um filtro analógico adaptativo em CMOS-VLSI, com recurso a técnicas de condensadores comutados e mecanismos de multiplexagem. A multiplexagem permite a reutilização de recursos, diminuindo desta forma o número de componentes no sistema. O trabalho a desenvolver passa pelo desenho e simulação do filtro e restantes blocos ao nível do transístor.

1.2 Motivação

Nos processos de transmissão, a informação emitida é afectada por ruído e distorção ao longo do canal de comunicação. De forma a maximizar a taxa de transmissão de informação, os receptores em geral possuem sistemas de igualização de canal, cujo objectivo é minimizar os efeitos de distorção e ruído do canal. Uma vez que a função que define o canal de transmissão normalmente não é conhecida, e dada a sua natureza não estacionária, isto é, o canal de transmissão é caracterizado por um sistema variante no tempo, a solução que em princípio melhor satisfaz o igualizador na recepção será a de um sistema adaptativo, sob a forma de um filtro ou até de uma rede neural artificial.

A implementação de um filtro analógico em CMOS-VLSI apresenta uma série de vantagens, entre elas, é eficiente em termos de área e potência. Normalmente, para o mesmo tipo de operações, os circuitos analógicos são mais compactos e mais rápidos. Para além da questão da realização, a resolução efectiva dos sistemas analógicos é infinita embora sujeito a ruído e aspectos não lineares que limitam a precisão. No entanto, como se tratam de sistemas adaptativos, em princípio a precisão deverá ser corrigida ao longo do processo de adaptação.

1.3 Estrutura da Dissertação

Para além da introdução, esta dissertação contém mais 5 capítulos. No capítulo 2 é realizada uma revisão bibliográfica sobre filtros adaptativos, é apresentado o filtro Gamma e o algoritmo LMS. No capítulo 3 estão expostas duas implementações em *Matlab/Simulink* de um filtro adaptativo, é referenciado o *software* utilizado e os circuitos pensados para a execução deste trabalho.

No capítulo 4 estão ilustradas as implementações do filtro multiplexado e de todos os blocos necessários ao sistema. Por fim, no capítulo 5 estão apresentadas as melhorias e trabalho futuro, bem como a conclusão desta tese.

Capítulo 2

Revisão Bibliográfica

Neste capítulo faz-se uma breve exposição sobre filtros adaptativos e a forma como se adaptam os seus coeficientes. Da mesma forma faz-se referência aos algoritmos mais usados e ao filtro que vai ser utilizado nesta implementação.

2.1 Sistemas Adaptativos

Como já foi referido, um filtro adaptativo é um sistema que autonomamente ajusta a relação entre o sinal de saída e um desejado. Este ajuste é feito de acordo com uma dada regra de adaptação que tem como objectivo minimizar a diferença entre o sinal de saída e um sinal desejado, recorrendo a um algoritmo adaptativo que determina o valor dos coeficientes do filtro. Assim, um filtro adaptativo pode ser resumido a dois blocos principais, um filtro linear e um algoritmo de adaptação. Como se pode observar na figura 2.1, o algoritmo possui como entradas $x(n)$ e $e(n)$ que no seu conjunto dão a informação necessária para adaptar os coeficientes do filtro.

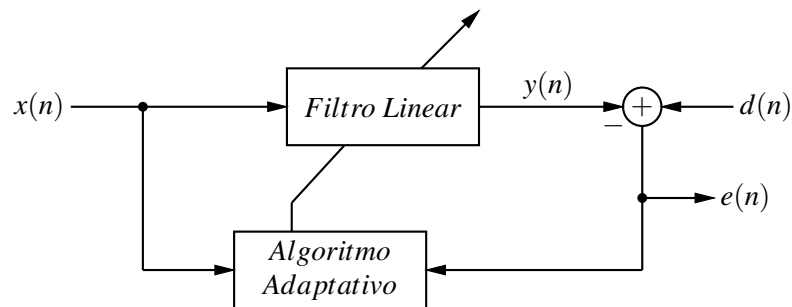


Figura 2.1: Blocos de um filtro adaptativo

As principais diferenças de um filtro adaptativo, face a um filtro dimensionado da forma tradicional, pode ser assim resumido.

Sinal	Descrição
$x(n)$	Entrada do filtro
$y(n)$	Saída do filtro
$d(n)$	Sinal desejado
$e(n)$	Erro obtido pela diferença de $d(n)$ por $y(n)$

Tabela 2.1: Entrada e saídas de um filtro adaptativo

- Um filtro é caracterizável por possuir uma entrada $x(n)$ e uma saída $y(n)$. Enquanto que um filtro adaptativo necessita de mais uma entrada para o sinal desejado $d(n)$ e produz mais uma saída, $e(n)$, que é dado pela diferença do sinal desejado e do sinal de saída, $d(n) - y(n)$.
- Os coeficientes de um filtro tradicional mantêm-se fixos, enquanto num filtro adaptativo os seus coeficientes variam de acordo com algoritmo utilizado, podendo eventualmente vir a fixarem-se no tempo, se entretanto o objectivo da função desempenho for atingido. [3]

Há, portanto, um conjunto de características que tornam os filtros adaptativos bastante atractivos para uma serie de aplicações práticas. Para além de outros e tomando como exemplo a maximização da relação sinal/ruído de um determinado sinal, este tipo de filtros consegue maximizar este parâmetro mesmo que a potência do ruído varie ao longo do tempo. Exige-se no entanto, um conjunto de sistemas auxiliares necessários ao processo de adaptação.

2.2 Implementações de Filtros Adaptativos

Nesta secção estão referenciadas algumas implementações CMOS analógicas de filtros adaptativos. Verifica-se que o número de publicações acerca deste tipo de implementações não é vasto. No entanto, a pesquisa apresentada de seguida foca unicamente os aspectos mais importantes para a realização deste trabalho. Estes resumem-se ao tipo de filtro, algoritmos, atrasos de sinal e aos factores que levaram os autores à escolha destes.

Um dos problemas recorrentes em muitas aplicações, como no cancelamento de eco, relaciona-se com a necessidade da utilização de um numero elevado de atrasos, alguns mais de 1000, para que o sistema funcione adequadamente. O artigo [4] apresenta várias estruturas para resolução deste problema. Discute-se aqui a questão do erro acumulado em filtros analógicos adaptativos, que surge com o uso filtros não ideais.

Outro problema nas implementações analógicas é o surgimento de defeitos na integração de circuitos e os *offsets* DC que degradam o desempenho. O artigo [5] que os *offsets* e a localização do pólo no circuito integrador, distorcem a variação dos parâmetros nos filtros analógicos adaptativos. Para testarem os efeitos enunciados recorreram ao algoritmo LMS e uma outra versão do mesmo algoritmo mas sem *offset* DC. Concluíram que os *offsets* introduzidos por atrasos, multiplicadores e outros circuitos, podem ser cancelados usando o algoritmos LMS, excepto se os *offsets* forem provocados pelas imperfeições nos circuitos.

No artigo [6] é discutido o problema da baixa sensibilidade dos circuitos não ideais. São apresentadas e analisadas estruturas de circuitos analógicos dinâmicos com boa performance, baseados em circuitos ressonantes. A simulação de um filtro adaptativo analógico baseado em técnicas de sintonia LC, é apresentado em [7], onde usam algoritmos DSL.

Em [8] é proposta a utilização de filtros adaptativos para corrigir a carga em sensores, que podem dar origem a respostas oscilatórias.

Em [9] os autores reportam a implementação de um *disturbance canceller* usando o algoritmo LMS. A implementação baseia-se na decomposição dos distúrbios dentro das componentes fase e quadratura do transformador de Hilbert.

No documento [10] é descrita a implementação de um filtro adaptativo para identificação de sistemas. Devido à impossibilidade de implementações analógicas de atrasos ideais utilizaram o filtro Gamma, recorrendo a um amplificador de transcondutância ligado como seguidor, com uma capacidade na saída para o transformar em circuito passa-baixo. A arquitetura do filtro Gamma permite que a rede adaptativa escolha os coeficientes do filtro de forma a otimizar a resolução do sistema. Verificaram que a resposta impulsional do filtro, com um número adequado de atrasos, aproxima-se de um atraso ideal. Finalmente demonstraram o conceito com um filtro adaptativo com três coeficientes, para realizar a identificação um circuito *Sallen-Key*.

Os autores dos documentos [11] e [12] implementaram uma metodologia para sintonizar sistemas analógicos usando circuitos translineares dinâmicos, recorrendo ao algoritmo LMS, numa topologia *fully differential*.

2.3 Algoritmos adaptativos

Os algoritmos adaptativos são caracterizados essencialmente pelos seguintes factores:

- Taxa de convergência
- Desajuste
- Complexidade numérica do algoritmo
- Estabilidade

A taxa de convergência é definida como o número de iterações que o algoritmo, perante sinais estacionários, necessita para se aproximar da solução óptima de Wiener-Hopf. O desajuste é o excesso de erro quadrático médio final relativamente ao valor do erro quadrático médio mínimo que um filtro de Wiener-Hopf produz. A complexidade numérica é o número de operações que o algoritmo necessita por cada iteração. O passo da iteração é controlado pela constante de adaptação, que simultaneamente determina a convergência do algoritmo.

Os algoritmos mais utilizados são os LMS e RLS. O RLS possui uma velocidade de convergência superior aos algoritmos de minimização do gradiente (LMS). No entanto a ordem de grandeza do algoritmo RLS é mais elevada, pois possui uma complexidade computacional muito superior à do algoritmo LMS, como evidencia a tabela 2.2. [13] [14]

Algoritmo	Multiplicações/Divisões	Somas/Subtrações	Total de operações
LMS	$2N$	$2N$	$4N + 3$
RLS	$3N^2 + 11N + 8$	$2N^2 + 6N + 4$	$5N^2 + 17N + 12$

Tabela 2.2: Complexidade numérica do LMS e do RLS

Uma vez que pretendemos implementar um filtro adaptativo em CMOS-VLSI, a escolha do algoritmo de adaptação cai sobre o LMS, pelas razões acima expostas.

2.4 Filtro Gamma

A escolha do filtro Gamma para implementações de aplicações adaptativas em CMOS-VLSI é adequada, pois são de fácil controlo e não apresentam atrasos ideais.

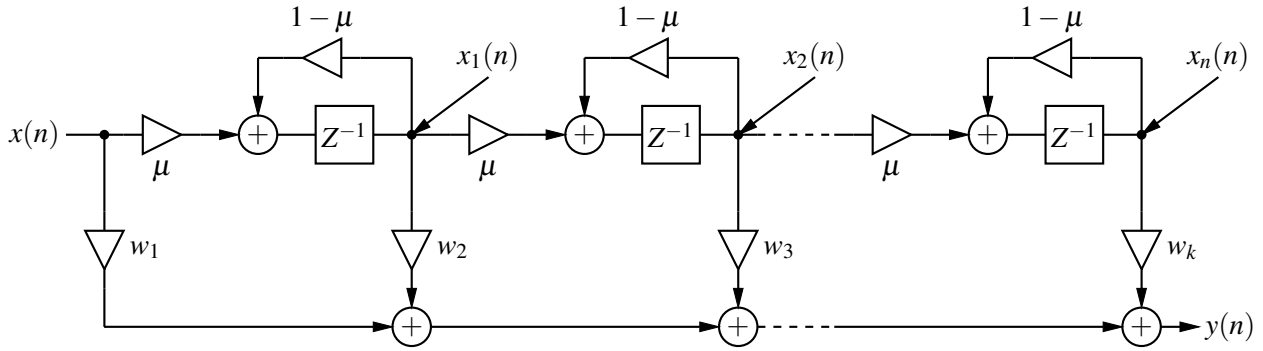


Figura 2.2: Estrutura de um filtro Gamma

A figura 2.2 representa a estrutura interna de um filtro Gamma, que é basicamente um conjunto de filtros em cascata. O filtro Gamma foi originalmente desenvolvido (de Vries and Principe, 1991) [15] para sistemas em tempo contínuo com aplicação em modelos de processamento de redes neurais. Os coeficientes do filtro Gamma são os coeficientes \vec{w}_k e a profundidade de memória que é representada por μ [16].

O filtro Gamma pode ser definido no tempo discreto como:

$$y(n) = \sum_{k=1}^K w_k x_k(n) \quad (2.1)$$

A função de transferência é dada por:

$$G(z) = \left(\frac{\mu}{z - (1 - \mu)} \right)^k \quad (2.2)$$

sendo k o número de *taps* utilizadas.

Observando a equação 2.2 facilmente se conclui que está garantida a estabilidade do filtro Gamma, desde que se mantenha o valor de μ entre 0 e 2 o que torna o controlo da estabilidade trivial, porque μ é comum a todos as *taps*.

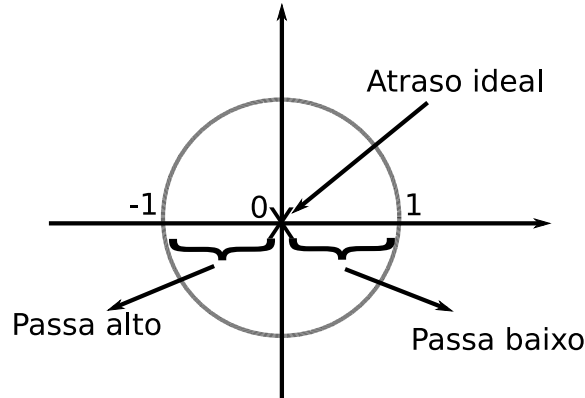


Figura 2.3: Comportamento do filtro Gamma para diferentes intervalos de μ .

Quando $\mu = 1$, a resposta impulsional reduz-se a um impulso unitário e transforma o sistema num filtro transversal adaptativo de Widrow. Se $\mu \neq 1$ a função de transferência implementa um atraso do tipo dispersivo [17].

2.5 Estrutura de um filtro adaptativo

Um filtro adaptativo pode ser caracterizado pela resolução que possui, que varia consoante o número de atrasos que o sinal de entrada sofre durante a filtragem. O sinal de saída $y(n)$ é representado pela soma pesada do sinal de entrada e dos vários atrasos que este sofre, como se pode visualizar na figura 2.4.

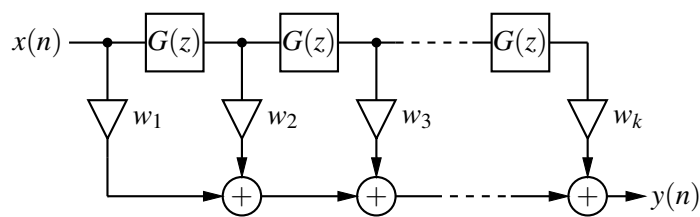


Figura 2.4: Estrutura de um filtro adaptativo

Para uma melhor compreensão do funcionamento do filtro adaptativo e de como se adaptam os coeficientes, é ilustrado de seguida as equações para a determinação destes.

Os coeficientes de um filtro de ordem k são dados por:

$$W(n) = [w_1(n), w_2(n), w_3(n), \dots, w_k(n)]^T \quad (2.3)$$

O sinal de saída do filtro é o somatório pesado dos vários atrasos, sendo estes dados por:

$$y(n) = \sum_{k=1}^K w_k(n)x_k(n) = W(n)^T X(n) \quad (2.4)$$

em que

$$X(n) = [x_1(n), x_2(n), x_3(n), \dots, x_k(n)] \quad (2.5)$$

O algoritmo LMS utiliza o valor instantâneo do erro quadrático $e^2(n)$, como estimativa da função de custo calculada pelo erro total do sistema. O erro é representado pela diferença do sinal desejado $d(n)$ e da saída do filtro $y(n)$. Como se vai ter possibilidade de observar, este algoritmo permite realizar a actualização dos coeficientes \vec{w}_k .

O erro do sistema é calculado por:

$$E = \sum_{n=0}^T E_n = \sum_{n=0}^T \frac{1}{2} e^2(n) = \sum_{n=0}^T \frac{1}{2} [d(n) - y(n)]^2 \quad (2.6)$$

A estimação do gradiente da função é dado por:

$$\vec{\nabla}(n) = \frac{\partial e^2(n)}{\partial w_k(n)} = -e(n)x_k(n) \quad (2.7)$$

Como o gradiente aponta no sentido da maximização da função de custo, desloca-se o vector dos coeficientes na direcção oposta, com o objectivo deste tender para o mínimo. Com isto temos que a variação dos coeficientes é dada por:

$$\Delta w_k = -\eta \frac{\partial E}{\partial w_k} \quad (2.8)$$

sendo η o passo de adaptação que controla a velocidade de convergência do algoritmo.

Desenvolvendo Δw_k temos:

$$\Delta w_k = -\eta \frac{\partial E}{\partial w_k} = \eta \sum_{t=0}^T e_t \frac{\partial E}{\partial w_k} = \eta \sum_{t=0}^T e_t x_k(n) \quad (2.9)$$

o que nos leva a dizer que a variação dos coeficientes pode ser aproximada por:

$$\Delta w_k(n) = \eta e(n)x_k(n) \quad (2.10)$$

As equações deduzidas foram retiradas do artigo [17].

Capítulo 3

Teste de funcionamento e blocos a implementar

Este capítulo apresenta um conjunto de experiências e ideias que representam a base que fundamenta o trabalho desenvolvido nesta dissertação. Foi implementado no *software Matlab/Simulink* um sistema de blocos para simular uma rede adaptativa. O uso desta ferramenta foi de certa forma útil, pois permitiu-nos rapidamente testar o funcionamento do filtro adaptativo. É ainda referenciado o *software* que foi utilizado para desenho e simulação do circuito em CMOS-VLSI.

Apresenta-se o filtro inicial e a dedução da sua função de transferência, bem como o circuito final que é necessário ser implementado. Após o conhecimento adquirido em sistemas adaptativos, através das expressões matemáticas ilustradas no capítulo anterior, foi possível projectar a interligação dos vários blocos para simular o funcionamento do filtro.

3.1 Simulação Matlab/Simulink com filtros FIR

O diagrama de blocos da figura 3.1 é composto por um filtro FIR adaptativo. O sinal de entrada é constituído por uma onda sinusoidal com 1V de amplitude e uma frequência de 50KHz, mais um bloco responsável por introdução do ruído. O filtro elaborado contém três atrasos, daí obtêm-se quatro coeficientes (\vec{w}_k).

Após 7000 iterações obteve-se os resultados apresentados nas figuras 3.2 e 3.3. Como se pode visualizar, temos um sinal sinusoidal de entrada $x(n)$ com ruído envolvente, o que eleva a amplitude do sinal de entrada do filtro para 2V, sendo o sinal desejado $d(n)$ de amplitude 1V com uma frequência de 50KHz. O sinal de saída do filtro $y(n)$ apresenta algum ruído, mas não tanto como o presente em $x(n)$. Com isto é possível afirmar que o filtro adaptativo conseguiu eliminar algum ruído, aproximando o valor da amplitude do sinal de saída do valor de amplitude do sinal desejado. Visualizando a evolução do erro $e(n)$, o algoritmo não consegue com estes blocos e

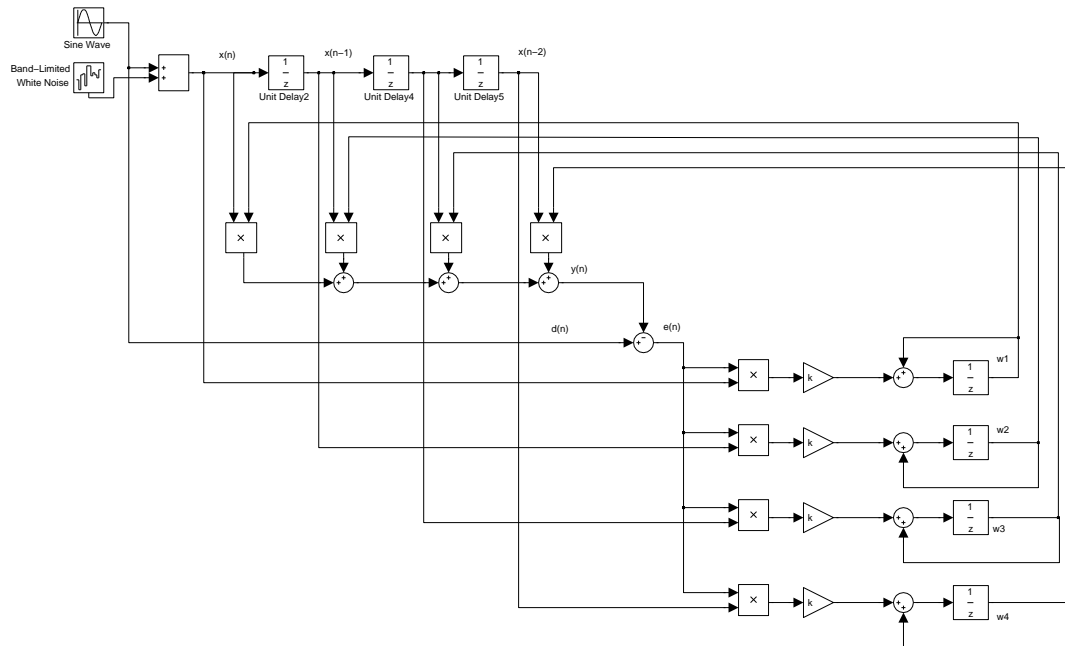


Figura 3.1: Filtro adaptativo com FIR's

valores atribuídos, minimizar mais a diferença entre o sinal desejado e o sinal de saída do filtro adaptativo.

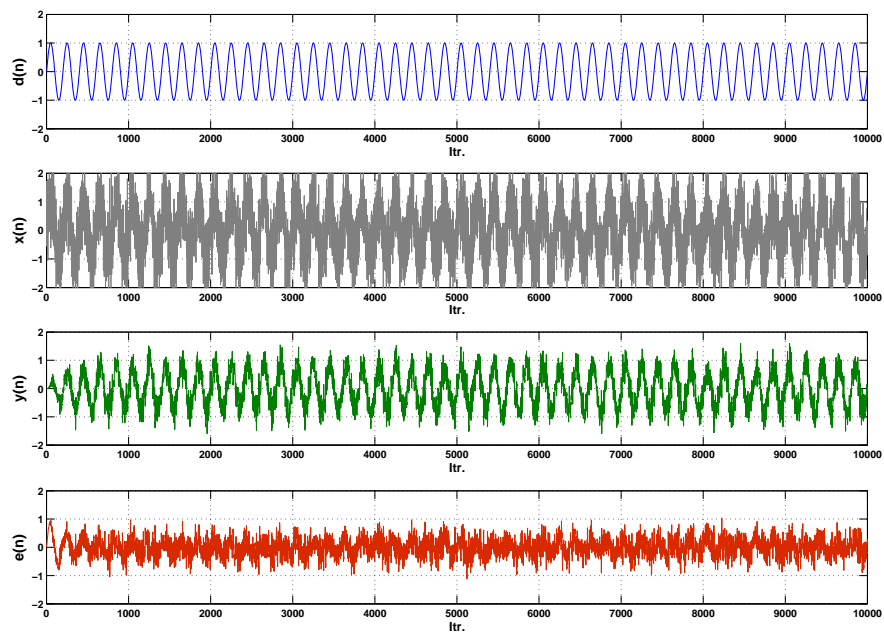


Figura 3.2: Resultados do filtro adaptativo com FIR's

Na figura 3.3 está ilustrada a evolução dos coeficientes que tendem todos para um valor equivalente.

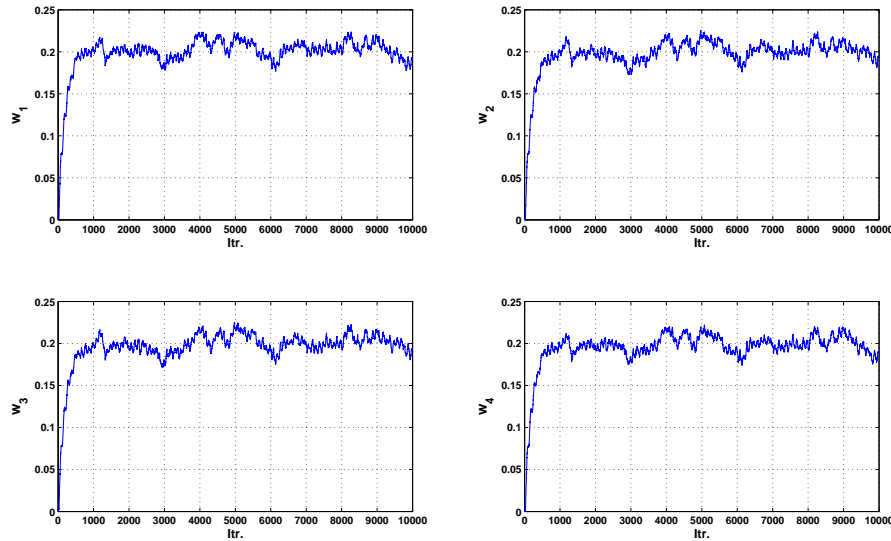


Figura 3.3: Coeficientes do filtro adaptativo com FIR's

Para obter-se melhores resultados, o número de atrasos do filtro FIR deveria aumentar, no entanto esta experiência serve apenas para comparar com o filtro Gamma.

3.2 Simulação Matlab/Simulink com filtro Gamma

Nesta simulação foi utilizado um filtro Gamma, como se pode visualizar em 3.4, é constituído por três *kernell* passa-baixo em cascata, havendo a necessidade de atribuir um valor a μ que controla a constante de tempo do filtro.

O valor de μ , nesta simulação, foi definido aleatoriamente, dentro de uma gama de valores para os quais se obtém um resposta tipicamente passa-baixo, concretamente com o valor $\frac{1}{3}$. Os restantes blocos são iguais aos utilizados na simulação 3.1, bem como os valores atribuídos aos blocos do circuito.

Comparando os resultados 3.5 e 3.2, é possível observar um melhoramento razoável somente com a substituição dos atrasos no filtros FIR pelo filtro Gamma. Com a utilização deste filtro, os valores finais da amplitude erro reduziram e observa-se que o sinal de saída possui menor ruído. Nesta simulação os valores dos coeficientes do sistema apresentados na figura 3.6 tendem para valores diferentes dos obtidos na simulação 3.1.

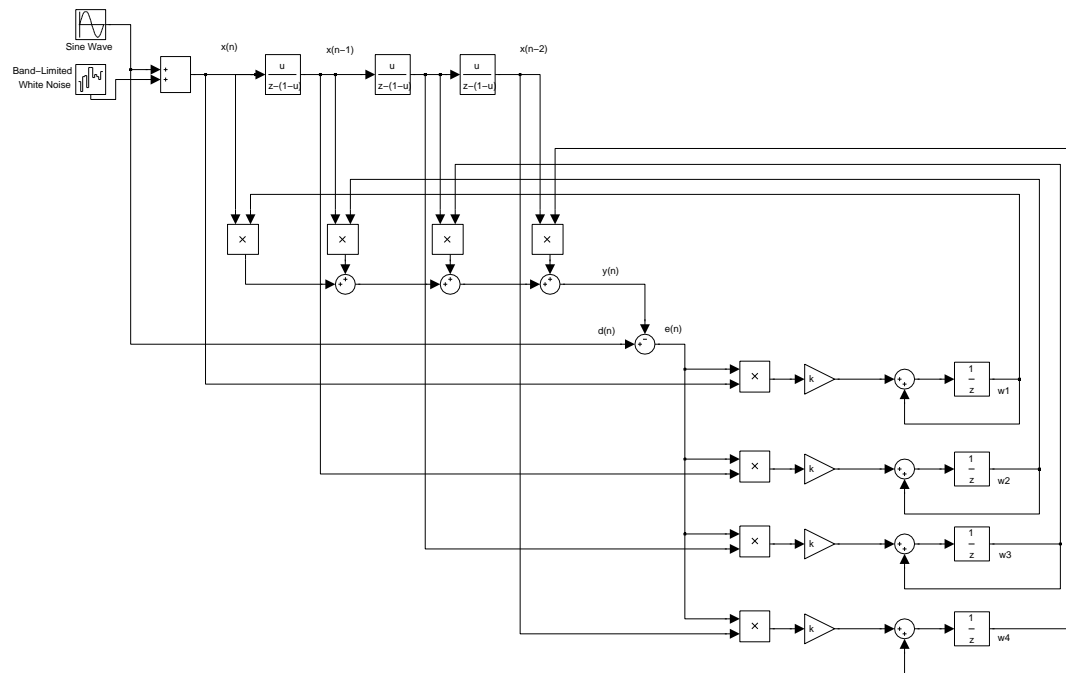


Figura 3.4: Filtro adaptativo com filtro Gamma

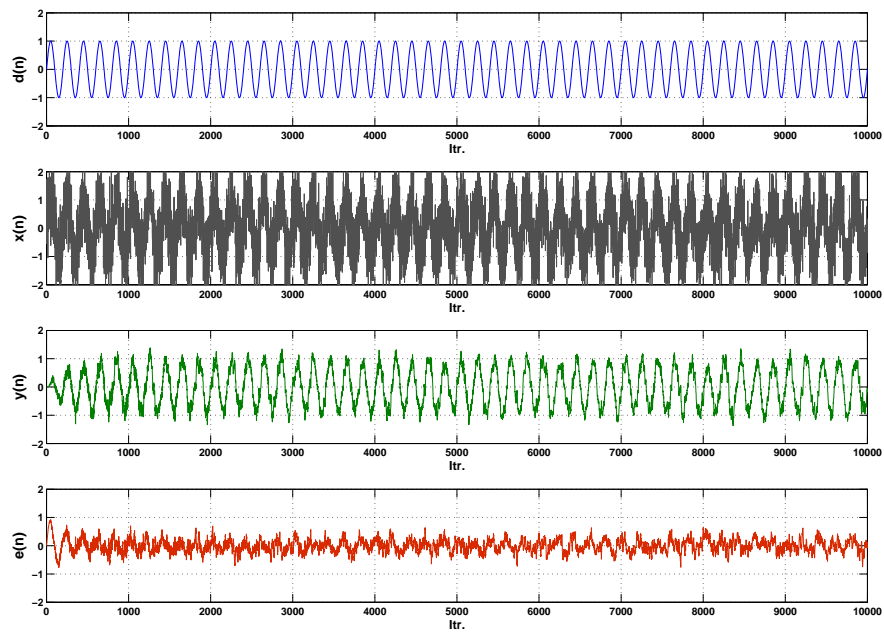


Figura 3.5: Resultados do filtro adaptativo com filtro Gamma

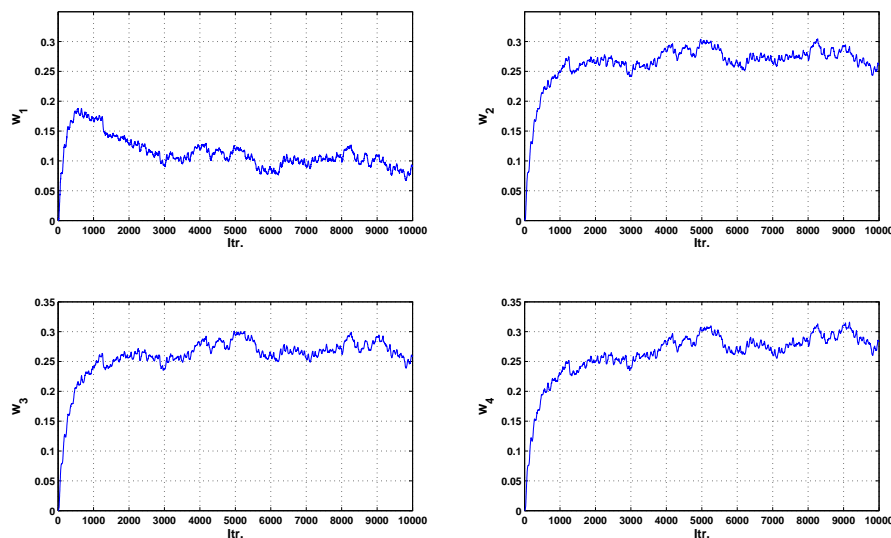


Figura 3.6: Coeficientes do filtro adaptativo com filtro Gamma

3.3 Ferramentas da tecnologia

A ferramenta de desenho e simulação do esquemático utilizada neste trabalho foi o *Cadence IC*. Esta ferramenta em junção com outras permite a implementação de circuitos digitais e analógicos, desde a fase de desenho até ao *tape-out* para produção.

Os esquemas dos circuitos apresentados nos capítulos seguintes foram editados na ferramenta *Virtuoso* que através da ferramenta *Analog Design Environment*, consegue funcionar com o *Cadence IC*. O *Analog Design Environment* é uma ferramenta de ambiente avançado de simulação. As implementações foram realizadas na tecnologia *CMOS-Standard 0.35μm* da *Austriamicrosystems*.

3.4 Filtro Passa-Baixo

Um filtro Gamma é um conjunto de filtros passa-baixo em cascata, logo é necessário implementar inicialmente o bloco passa-baixo para verificar a sua resposta. A implementação deste filtro, recorrendo a técnicas de circuito de condensadores comutados, encontra-se ilustrado na figura 3.7

O filtro é controlado por dois sinais não sobrepostos, ϕ_1 e ϕ_2 . Considerando que ϕ_1 acontece primeiro que ϕ_2 , a carga do circuito em ϕ_1 é dada por:

$$-V_{in}[(n-1)T]C_1 - V_{out}[(n-1)T]C_\mu \quad (3.1)$$

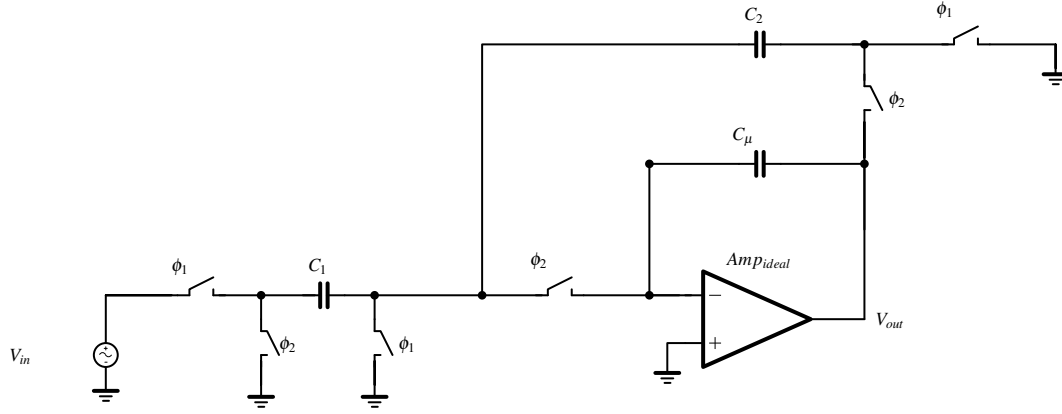


Figura 3.7: Filtro passa-baixo

No instante ϕ_2 a carga do circuito é:

$$-V_{out}[(n-1)T](C_2 + C_\mu) \quad (3.2)$$

Assumindo que a carga nos dois instantes é a mesma, temos:

$$-V_{in}[(n-1)T]C_1 - V_{out}[(n-1)T]C_\mu = -V_{out}[(n-1)T](C_2 + C_\mu) \quad (3.3)$$

Com a equação às diferenças, é possível calcular a função de transferência do filtro:

$$-C_1 V_{in}(Z)Z^{-1} - C_\mu V_{out}(Z)Z^{-1} = (C_2 + C_\mu)V_{out}(Z) \quad (3.4)$$

$$C_1 V_{in}(Z)Z^{-1} = (C_2 + C_\mu - C_\mu Z^{-1})V_{out}(Z) \quad (3.5)$$

$$\frac{V_{out}(Z)}{V_{in}(Z)} = \frac{C_1 Z^{-1}}{C_2 + C_\mu - C_\mu Z^{-1}} = \frac{C_1}{(C_2 + C_\mu)Z - C_\mu} \quad (3.6)$$

$$\frac{V_{out}(Z)}{V_{in}(Z)} = \frac{\frac{C_1}{C_2 + C_\mu}}{Z - \frac{C_\mu}{C_2 + C_\mu}} \quad (3.7)$$

Considerando que C_1 é igual a C_2 temos:

$$\frac{V_{out}(Z)}{V_{in}(Z)} = \frac{\frac{C_1}{C_1 + C_\mu}}{Z - \frac{C_\mu}{C_1 + C_\mu}} \quad (3.8)$$

Substituindo $\frac{C_1}{C_1+C_\mu}$ por μ temos a função de transferência dada por:

$$G(Z) = \frac{\mu}{Z - (1 - \mu)} \quad (3.9)$$

As capacidades C_μ e C_1 permitem definir o valor de μ da função de transferência. É com o sinal ϕ_1 que se faz a amostragem.

Os sinais de controlo referidos em cima estão representados na figura 3.8. Como se pode visualizar, ϕ_1 e ϕ_2 estão em oposição de fase e com transições não sobrepostas. Caso não se verifiquem estes requisitos, o filtro não iria funcionar correctamente. Estes sinais de controlo foram criados recorrendo a uma fonte de ondas quadradas, existente na ferramenta usada, com os seguintes parâmetros:

Sinal	Período	Atraso	Tempo de Impulso	Valor do Impulso
ϕ_1	2000ns	0ns	950ns	1.65V
ϕ_2	2000ns	1000ns	950ns	1.65V

Tabela 3.1: Sinais de controlo ϕ_1 e ϕ_2

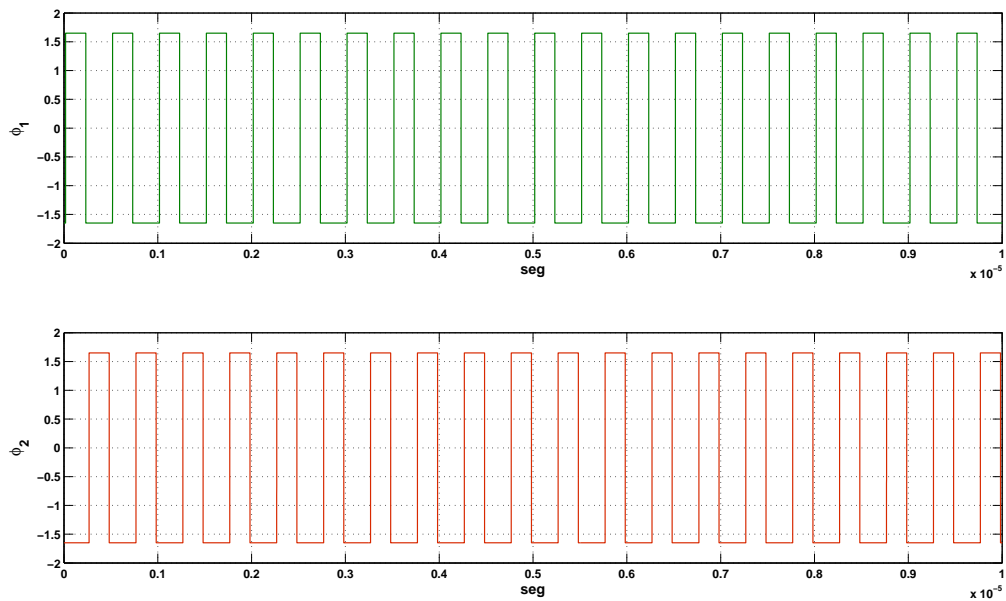


Figura 3.8: Sinais de controlo ϕ_1 e ϕ_2

Neste circuito existe ainda um amplificador, com um ganho elevado (10000V/V) e como os valores da saturação definidos a +2V e a -2V. Os *switchs* comutam para fecho para sinais de controlo acima dos 100mV e aberto para sinais abaixo dos -100mV.

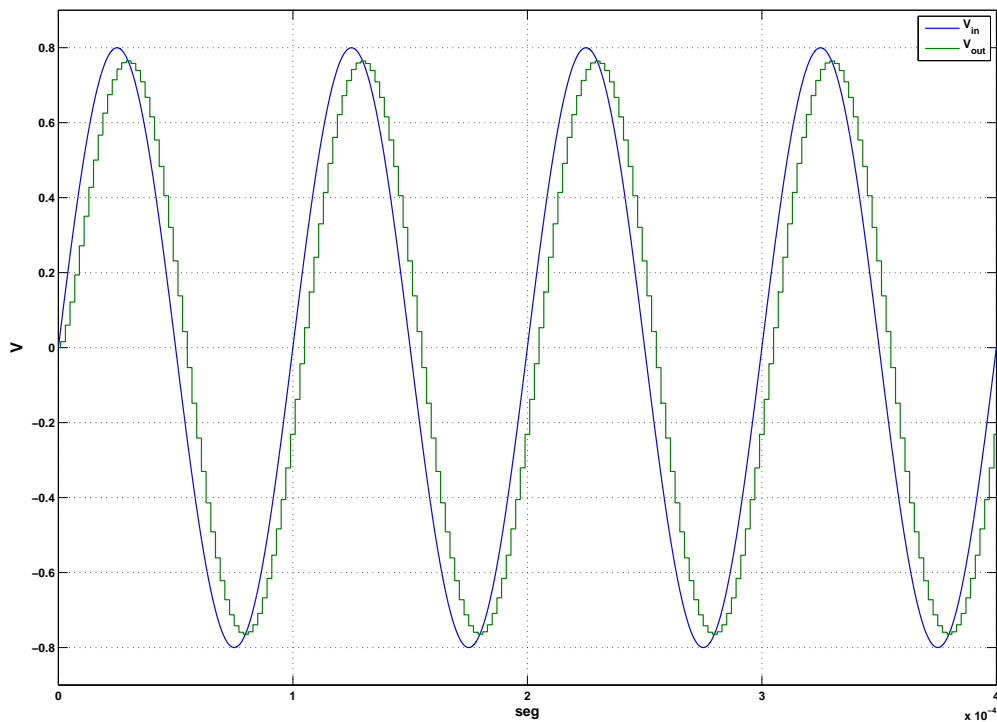


Figura 3.9: Resposta filtro passa-baixo

O valor atribuído às capacidades C_1 e C_2 na simulação foi $1pF$ para definir ganho unitário no filtro, e a capacidade C_μ foi atribuído $2pF$ para que o valor de μ estar definido com $\frac{1}{3}$. Na entrada V_{in} foi aplicado um sinal sinusoidal com uma amplitude de $800mV$ e com frequência de $10KHz$, foi gerado na saída a forma de onda ilustrada na figura 3.9.

3.5 Circuito a implementar

O circuito da figura 3.10 e 3.11 ilustram os esquemas de partida para implementação dos blocos necessários. Estes circuitos implementam uma metodologia de multiplexagem, que permite gerar, neste caso, três filtros usando a estrutura de apenas um e partilhando um único multiplicador para todo o sistema adaptativo, minimizando deste modo os recursos. A multiplexagem é conseguida através do uso de *switches*, que ao comutarem alteram as ligações do circuito.

Na figura 3.10 está integrado o filtro passa-baixo apresentado na secção anterior. Este circuito começa por amostrar o sinal de entrada e coloca-o na entrada do filtro no instante ϕ_1 . A integração é realizada no instante ϕ_2 e o que resulta é guardado no mesmo instante. Quando aparecer o próximo ϕ_1 este último sinal é introduzido na entrada do filtro, obtendo um novo atraso em ϕ_2 que é também armazenado. De forma resumida, ϕ_1 realiza a amostragem e ϕ_2 a integração. Este processo pode ser repetido várias vezes, tantos quantos atrasos pretendermos realizar, havendo unicamente a necessidade de colocar o número de condensadores igual ao número de atrasos, para estes serem memorizados. O que permite aos *switch's* comutarem são os sinais de relógio ϕ_1 e ϕ_2 ,

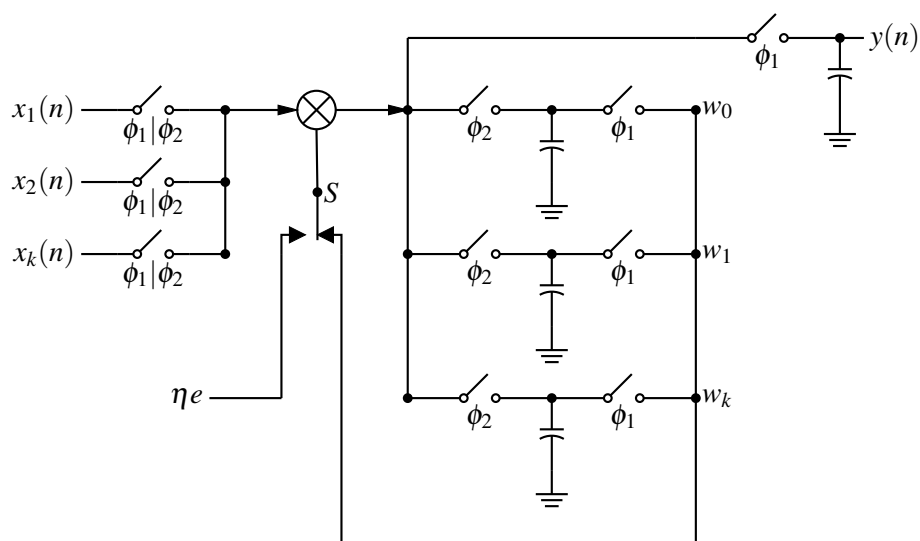


Figura 3.11: Esquema inicial para calculo da saída e coeficientes

capacidade de saída $y(n)$. A outra equação necessária é a actualização dos coeficientes, calculada da seguinte forma:

$$\Delta w_k(n) = \eta e(n) x_k(n) \quad (3.11)$$

Colocando o *switch* na outra posição, o valor $\eta e(n)$ num instante ϕ_2 é multiplicado pelo valor que está nas entradas $x_k(n)$, sendo o resultado da multiplicação colocado nas capacidades, guardando o valor dos coeficientes.

Capítulo 4

Implementações e Resultados

Neste capítulo são demonstrados todos os passos da implementação do filtro adaptativo no software anunciado em 3.3. Com base no filtro passa-baixo da secção anterior foram implementado três filtros em cascata para serem comparados com um filtro multiplexado de três atrasos. Além do filtro a ser usado no sistema, também é necessário implementar outros blocos que permitam realizar os cálculos da rede adaptativa. No final é ilustrado o circuito final com os respectivos resultados.

4.1 Cascata de Filtros Passa-Baixo

Antes de avançarmos para um filtro multiplexado, é necessário a simulação de um filtro não multiplexado, ou seja uma série de filtros. Só desta forma é possível afirmarmos posteriormente que a multiplexagem funciona correctamente, comparando os seus resultados com estes. Deste modo foram simulados três filtros em cascata de acordo com o circuito de apresentado na figura 4.1, sendo o esquema de cada filtro igual ao filtro apresentado na secção anterior.

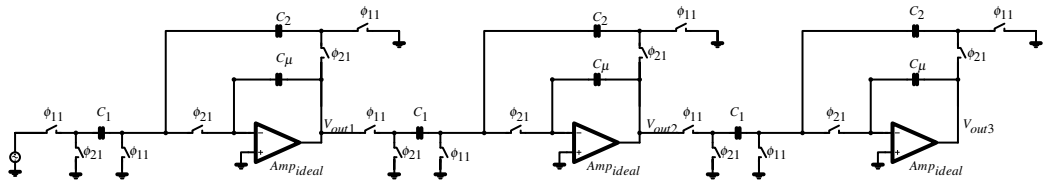


Figura 4.1: Série de três filtros passa-baixo

A simulação do circuito apresentado nesta secção foi efectuada com os valores das capacidades C_1, C_2 e C_μ iguais aqueles atribuídos na simulação da secção 3.4. Os valores que caracterizam o amplificador e os *switches* são de igual forma os mesmos apresentados anteriormente. A figura 4.2 apresenta os resultados da simulação realizada.

Os sinais de controlo utilizados nesta simulação, como na seguinte, estão ilustrados na figura 4.3 e os seus valores na tabela 4.1.

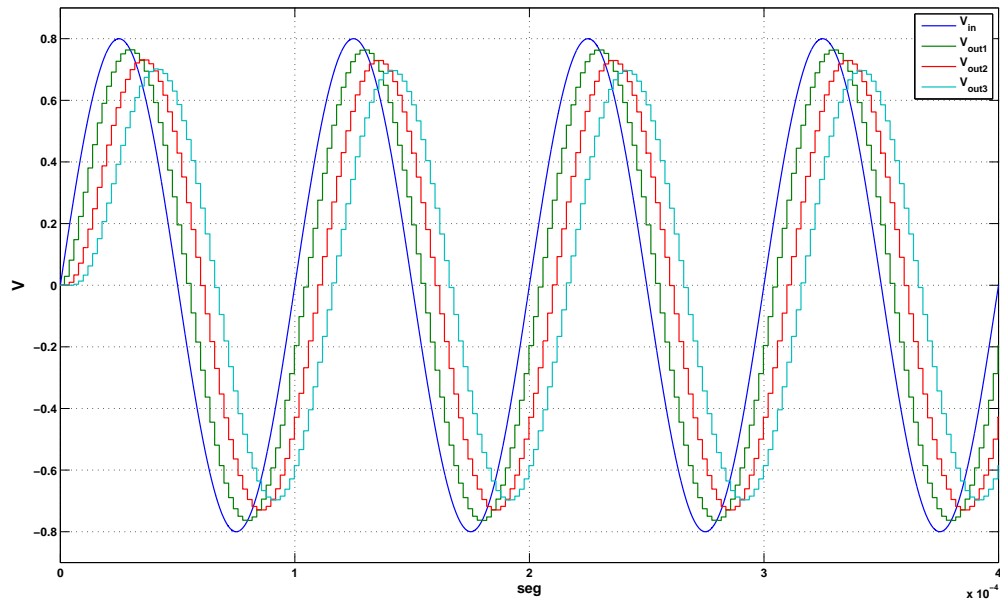


Figura 4.2: Cascata de filtros

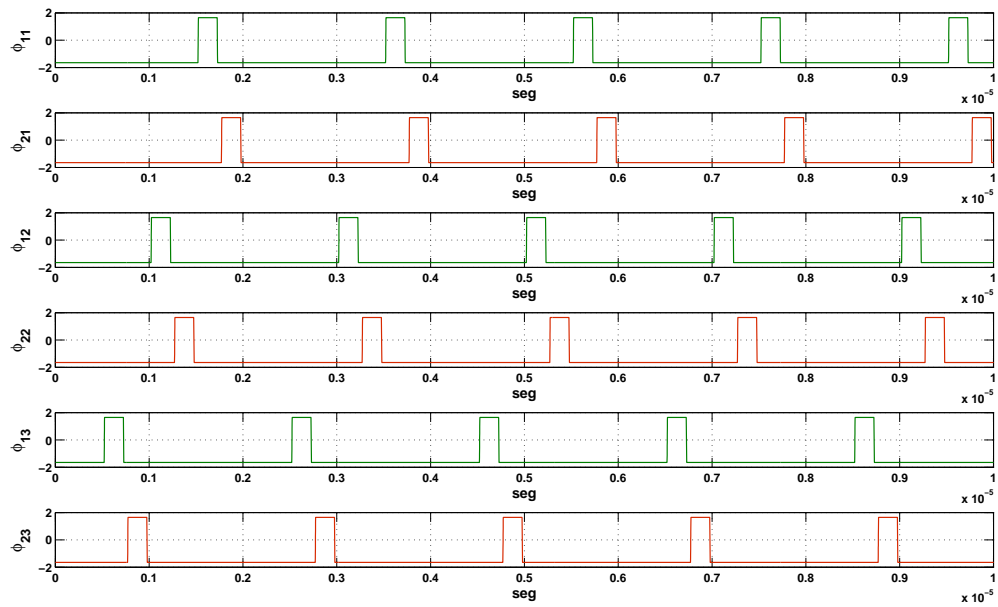


Figura 4.3: Sinais de controlo para filtro multiplexado

Sinal	Período	Atraso	Tempo de Impulso	Valor do Impulso
ϕ_{11}	1500ns	1020ns	200ns	1.65
ϕ_{21}	1500ns	1270ns	200ns	1.65
ϕ_{12}	1500ns	520ns	200ns	1.65
ϕ_{22}	1500ns	770ns	200ns	1.65
ϕ_{13}	1500ns	20ns	200ns	1.65
ϕ_{23}	1500ns	270ns	200ns	1.65

Tabela 4.1: Sinais de controlo para série de Filtros

4.2 Filtro multiplexado

A forma idealizada para implementar um filtro multiplexado encontra-se ilustrada na figura 4.4. Comparando esta arquitectura com a descrita na secção anterior, verifica-se agora que o sinal de saída do filtro, em vez de se deslocar para a entrada do seguinte, é armazenado num elemento de memória capacitivo para reutilizar mais tarde.

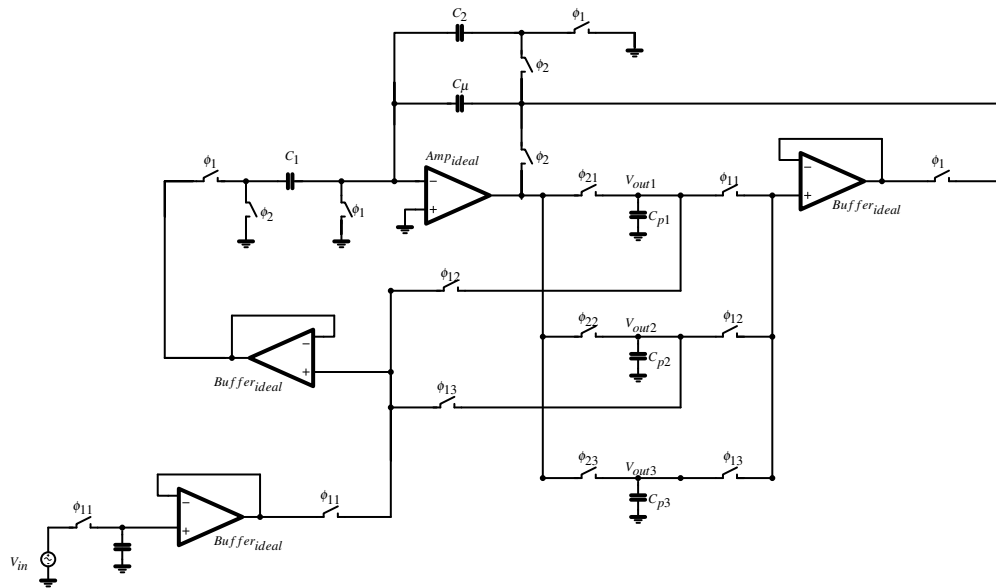


Figura 4.4: Três filtros multiplexados

Como não é possível, usando multiplexagem, realizar a amostragem do sinal de entrada e a filtragem com dois únicos sinais de controlo, os sinais ϕ_1 e ϕ_2 deram origem a outros sinais representados na figura 4.3. Como estes sinais foram criados à imagem de ϕ_1 e ϕ_2 , não existe qualquer sobreposição entre eles.

O sinal de entrada é amostrado em ϕ_{11} e é integrado logo de seguida em ϕ_{21} . No instante seguinte ϕ_{12} , a saída da primeira filtragem armazenada na capacidade C_{p1} , é colocada na entrada do filtro, sendo a filtragem realizada em ϕ_{22} e o seu valor resultante guardado na capacidade C_{p2} . Este processo volta a repetir-se com os sinais de controlo seguintes e o valor da nova filtragem é armazenado na capacidade C_{p3} . Os amplificadores/buffers ideais têm as mesmas características do

amplificador descrito no capítulo anterior. Devido ao processo de multiplexagem, é necessário a preservação da carga nas capacidades enquanto estas não estão a ser necessárias, logo o uso dos *buffers* é fulcral para isolar a partilha de carga entre as capacidades. Reparar que havendo necessidade de mais atrasos, basta adicionar capacidades para armazenar os coeficientes e os respectivos *switches*.

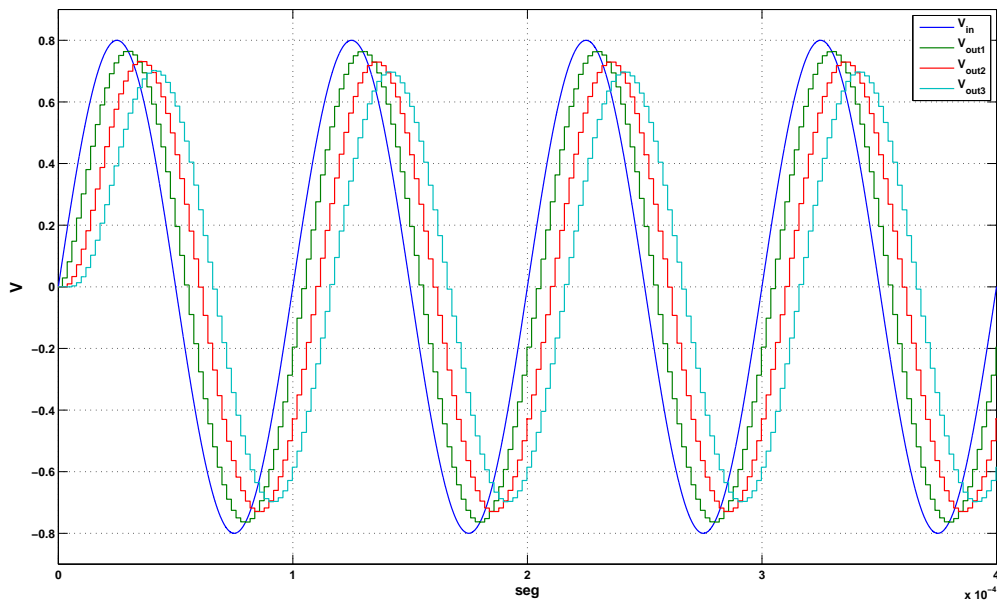


Figura 4.5: Resposta filtro multiplexado

Comparando os sinais obtidos com os da figura 4.2, é possível afirmar que o filtro multiplexado funciona de igual forma a uma série de filtros.

4.3 Offset

O filtro multiplexado possui a capacidade de compensar os *offsets* existentes no sistema de filtragem, o que é importante nos sistemas adaptativos. Para se verificar que assim é introduziu-se um offset de $500mV$ nos amplificadores. Como se pode visualizar na figura 4.6 os sinais filtrados começam próximos do valor de *offset*, mas rapidamente o filtro consegue corrigir esse valor.

4.4 Análise de operacionalidade

Como já foi referido, o filtro multiplexado possui a mesma resposta que vários filtros em série. Por cada atraso que seja necessário acrescentar, basta unicamente adicionar uma capacidade para memorizar o seu valor e o respectivo *switch*. Para implementação do filtro multiplexado é necessário ter em atenção a resistência interna dos *switches* enquanto estes estão em aberto. Esta

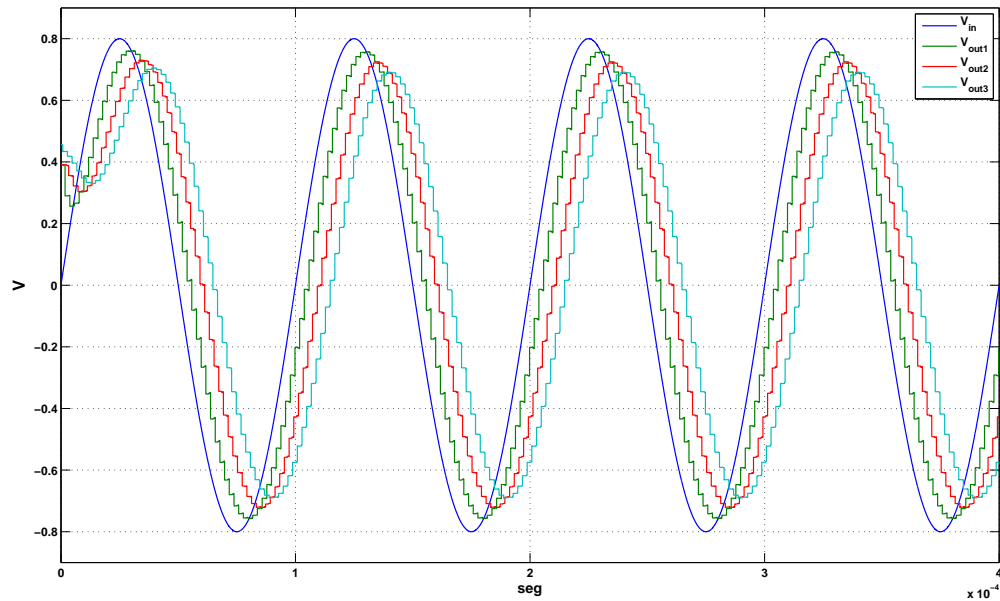


Figura 4.6: Resposta filtro multiplexado com *offset*

resistência tem de ser extremamente elevada para impedir a passagem de carga armazenada nas capacidades dos coeficientes para outras capacidades.

4.5 Outros blocos necessários

Após a verificação do filtro multiplexado, é necessária a implementação dos outros blocos do filtro adaptativo. Observando as equações 2.4, 2.6 e 2.10, verifica-se ser necessário implementar um somador para soma dos atrasos pesados do filtro, um subtrator para cálculo do erro e um integrador para actualizar o valor dos coeficientes. Um outro bloco que vai ser utilizado várias vezes é o amplificador que também irá ser implementado.

4.5.1 Somador - Somatório pesado dos coeficientes

Este somador é o responsável por realizar a soma pesada dos vários coeficientes do filtro. A equação que tem que realizar é a seguinte:

$$y(n) = \sum_{k=0}^K w_k x_k(n) \quad (4.1)$$

Neste caso foi implementado um somador de duas entradas, mas pode facilmente ser alterado para um número indiferenciado de entradas.

Os sinais de entrada são amostrados nas capacidades C_1 e C_2 , nos respectivos instantes ϕ_{1k} sendo depois integrados individualmente, em ϕ_{2k} , para a capacidade C_3 que vai acumular a soma,

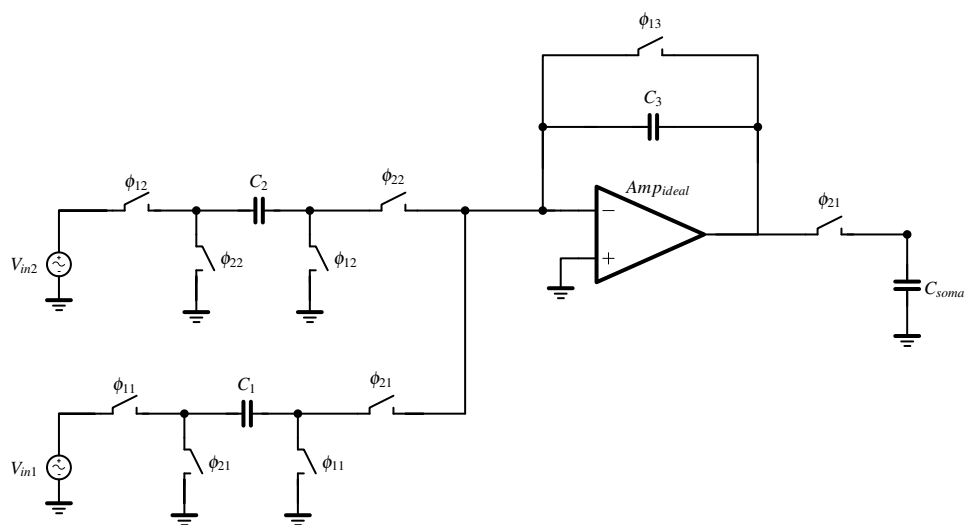


Figura 4.7: Somador duas entradas

enviando o resultado final, em ϕ_{21} , para a capacidade C_{soma} . Quando o valor final da soma for memorizado na capacidade C_{soma} , a capacidade C_3 é "limpa", preparando-se para a efectuar um nova soma no ciclo seguinte.

Para testar o somador foram colocadas nas entradas ondas sinusoidais, uma com $600mV$ e outra com $400mV$ de pico, a uma frequência de $10KHz$. Espera-se na saída uma onda sinusoidal com $1V$ de amplitude e com a mesma frequência.

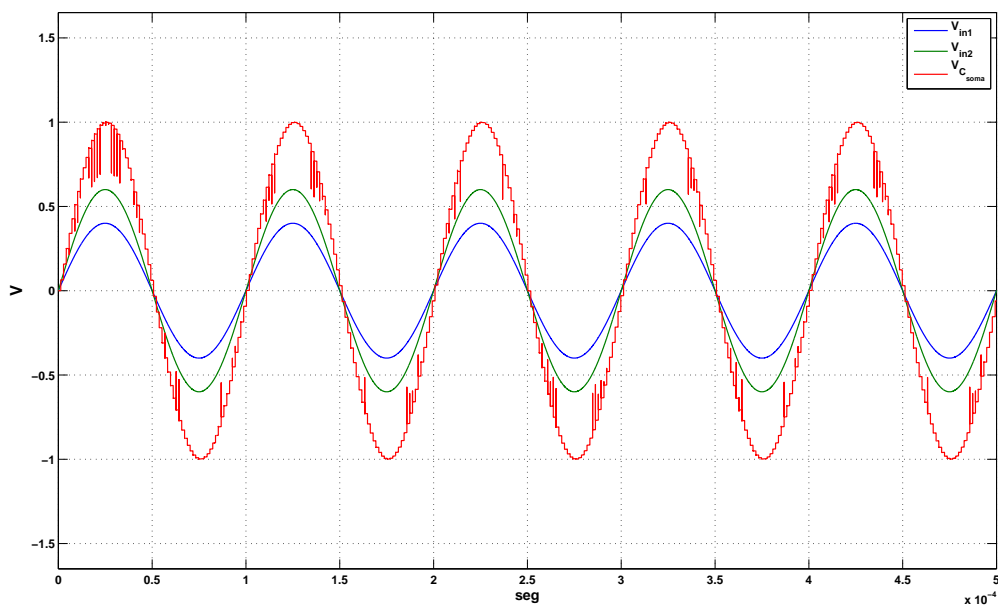


Figura 4.8: Resultados do somador dos atrasos do filtro

Estes valores comprovaram-se como ilustra a figura 4.8, pois o valor na capacidade C_{soma}

contém a soma perfeita dos sinais de entrada. Os sinais de controlo usados estão ilustrados na figura 4.9.

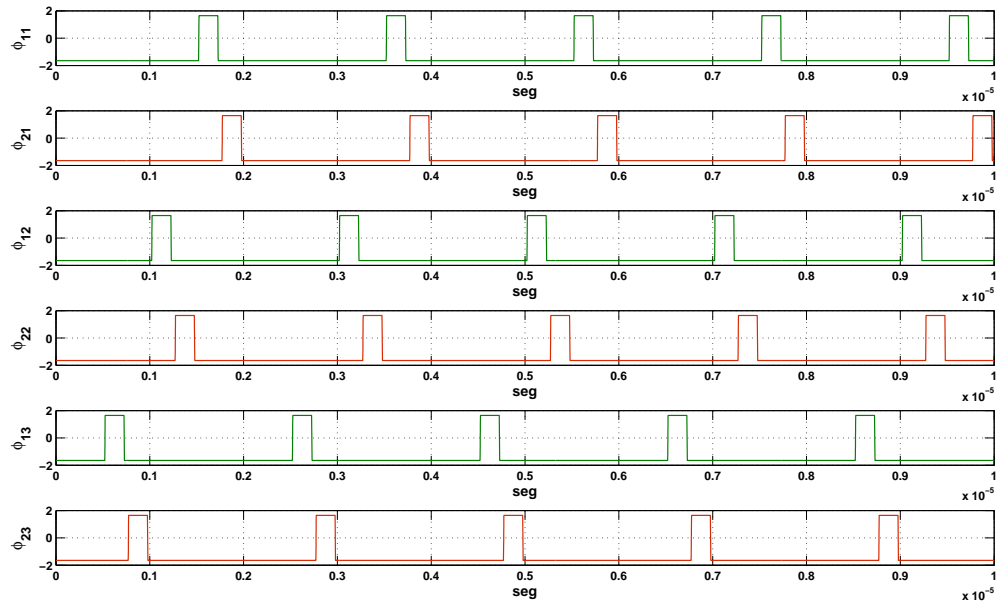


Figura 4.9: Sinais de controlo para somador dos atrasos do filtro

4.5.2 Integrador - Actualização dos coeficientes

Para o cálculo de um novo coeficiente $w_k(n)$ é necessário o coeficiente no instante anterior $w_k(n-1)$. O cálculo dos coeficientes é dado por:

$$w_k(n) = w_k(n-1) + \eta e(n)x_k(n) \quad (4.2)$$

O circuito pensado para efectuar este cálculo baseia-se num integrador, como ilustra a figura 4.10. Os coeficientes ficam sempre armazenados nas capacidades de realimentação do amplificador, neste caso C_{s1} e C_{s2} . Por exemplo, o valor de um coeficiente é amostrado na capacidade C_1 no instante ϕ_{11} e depois este valor será acumulado ao valor armazenado na capacidade C_{s1} no ϕ_2 .

Este bloco foi testado, introduzindo nas fontes V_1 e V_2 , $50mV$ e $20mV$ contínuos, respectivamente. Como se pode visualizar nos resultados obtidos na figura 4.11, a acumulação constante nas capacidades realiza-se correctamente, os valores dos saltos da tensão são praticamente lineares. Os sinais de controlo são os mesmos da secção anterior.

Com esta arquitectura o problema da perda de carga não é visível, pois os condensadores que armazenam os valores da soma estão ligados com os *switches* a pontos de baixa impedância ou nós de 0V.

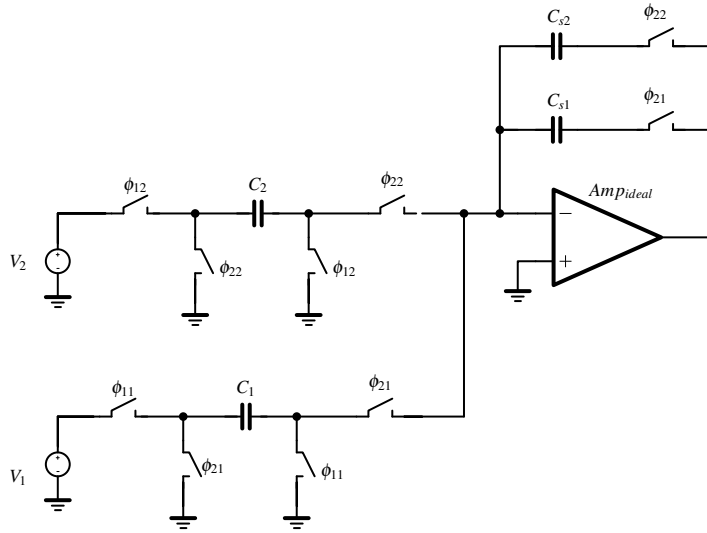


Figura 4.10: Integrador para actualização dos coeficientes

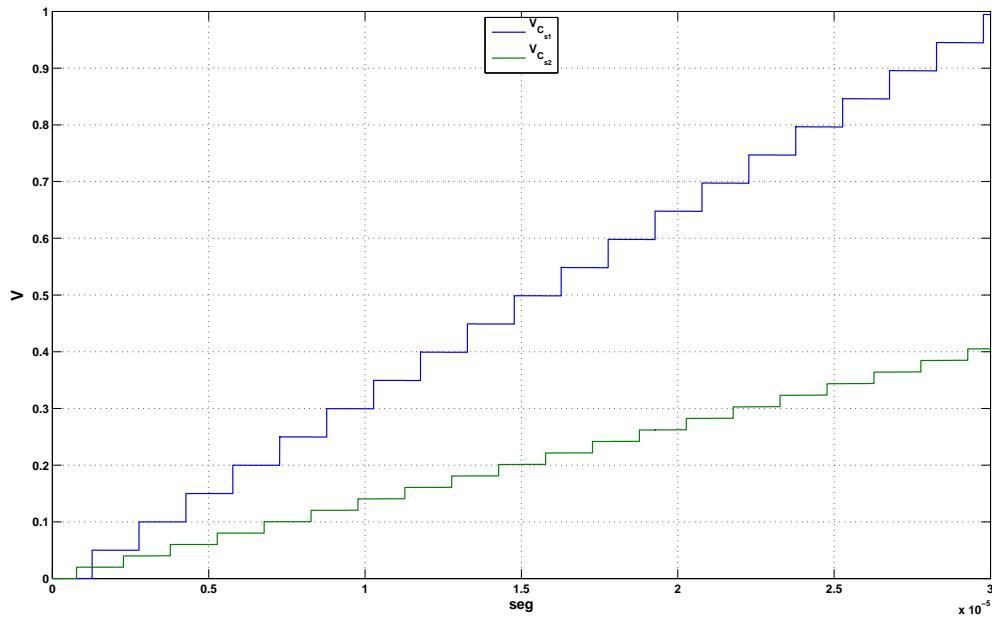


Figura 4.11: Resultados do integrador para actualização dos coeficientes

4.5.3 Subtractor - Cálculo do erro

O subtractor apresentado na figura 4.12 é o responsável por calcular o erro pesado $\eta e(n)$ entre o sinal desejado $d(n)$ e o soma pesada do filtro $y(n)$.

O subtractor está a colocar o valor da tensão que tem na fonte V_{in1} na capacidade C_e durante ϕ_{25} . No instante ϕ_{15} é injectado, pelo outro pólo do condensador, o valor que está em V_{in2} -sinal desejado. A tensão memorizada inicialmente na capacidade vai ser somada negativamente, pois

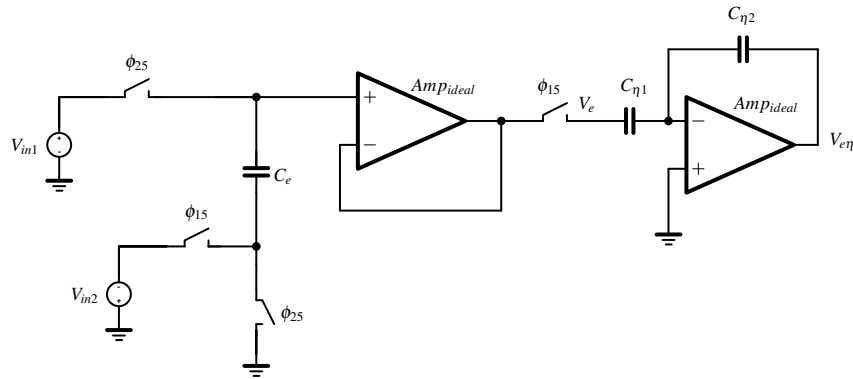


Figura 4.12: Subtractor para cálculo do erro

V_{in2} está em oposição de fase em relação a V_{in1} . Após a obtenção da subtração, o valor gerado passa por um amplificador para mudar o valor de amplitude consoante o ganho definido por $\frac{C_{\eta1}}{C_{\eta2}}$, definindo deste modo o valor de η . Notar que V_{in1} é o sinal $y(n)$ e V_{in2} é o sinal $d(n)$

Para simular este somador, V_{in1} e a V_{in2} são ondas sinusoidais de frequência $10kHz$ com $600mV$ e $800mV$ de amplitude respectivamente.

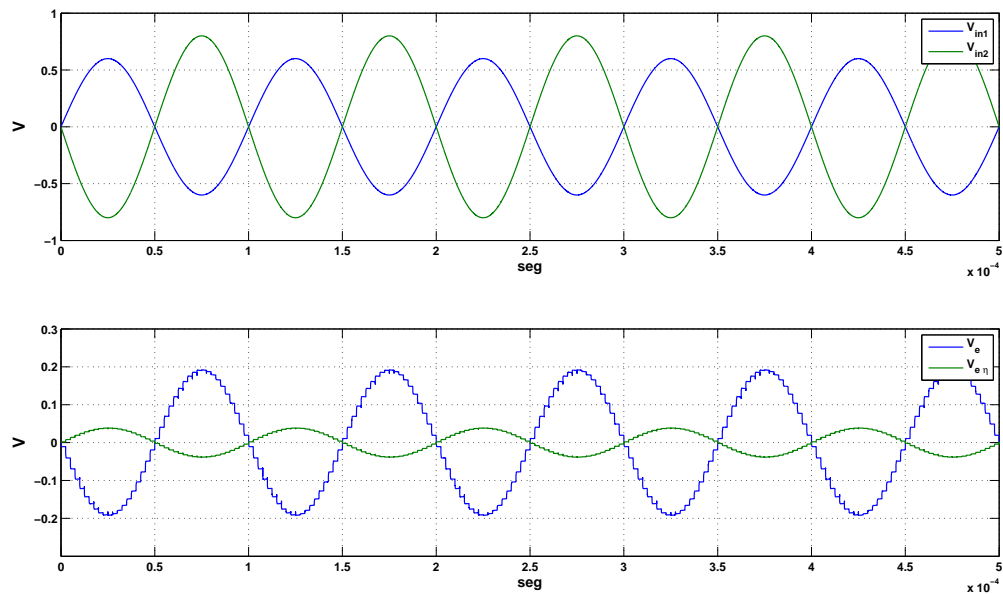


Figura 4.13: Resultados do subtractor para cálculo do erro

Como se pode visualizar na figura 4.13, os sinais V_{in1} e V_{in2} estão em oposição de fase, geram um sinal V_e com amplitude de $200mV$, resultantes da operação $-(V_{in1} + V_{in2})$. O sinal V_e é depois atenuado e invertido através das capacidades $C_{\eta1}$ com $2pF$ e $C_{\eta2}$ com $10pF$, resultando no sinal $V_{e\eta}$.

Os sinais de controlo usados estão ilustrados na figura 4.14, estando ϕ_{25} quase o tempo todo

a 1.65V. Os dois sinais descritos anteriormente estão em oposição de fase e não possuem sobreposições nas transições entre eles.

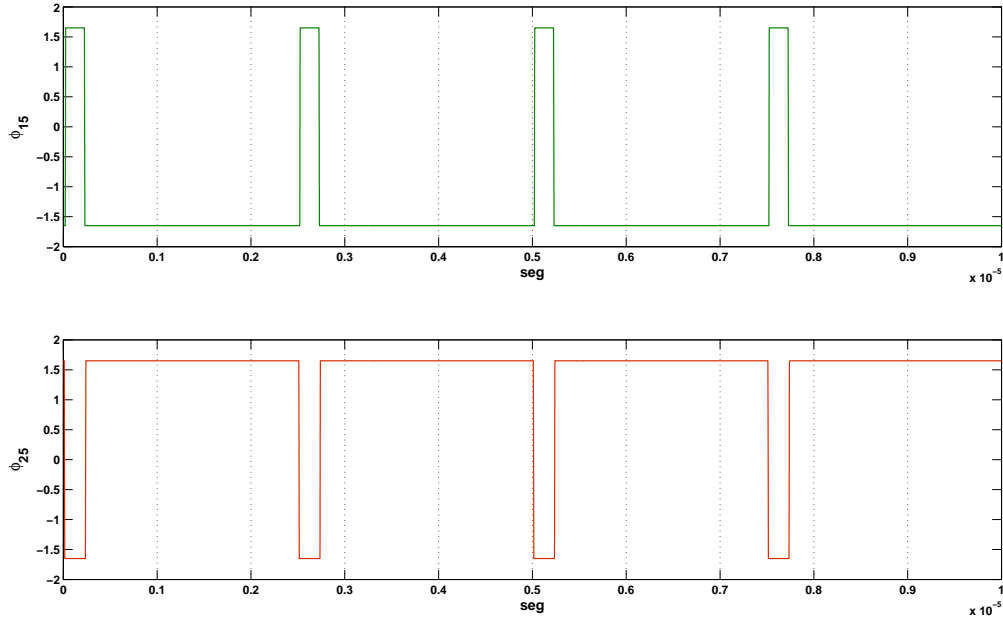


Figura 4.14: Sinais de controlo para subtrator de cálculo do erro

4.6 Estrutura final

Após o desenho de todos os blocos necessários para implementar um filtro adaptativo, foi estruturado o esquema apresentado na figura 4.15 composto por componentes ideais. Este circuito está preparado para realizar uma filtragem adaptativa utilizando dois coeficientes.

Os sinais de controlo deste circuito estão ilustrados na figura 4.16 e foram gerados conforme os valores apresentados na tabela 4.2. Notar que os sinais ϕ_1 e ϕ_2 são criados a partir de ϕ_{11} , ϕ_{12} , ϕ_{21} e ϕ_{22} , pois ϕ_1 e ϕ_2 só estão a ser usados no filtro multiplexado.

Como se pode visualizar no circuito, o sinal de entrada e o sinal de saída dos filtros estão disponíveis no mesmo ponto em ϕ_{11} e ϕ_{12} respectivamente. Estes sinais, $x_k(n)$, vão ser multiplicados individualmente no instante ϕ_1 pelos valores armazenados nas capacidades C_{s1} e C_{s2} , que correspondem aos valores dos coeficientes $w_k(n)$ da equação:

$$y_k(n) = w_k(n)x_k(n) \quad (4.3)$$

Após estas multiplicações, os valores resultantes $y_k(n)$, vão ser somados pelo somador apresentado na secção 4.5.1, para calcular o somatório pesado dos vários atrasos como indica a equação

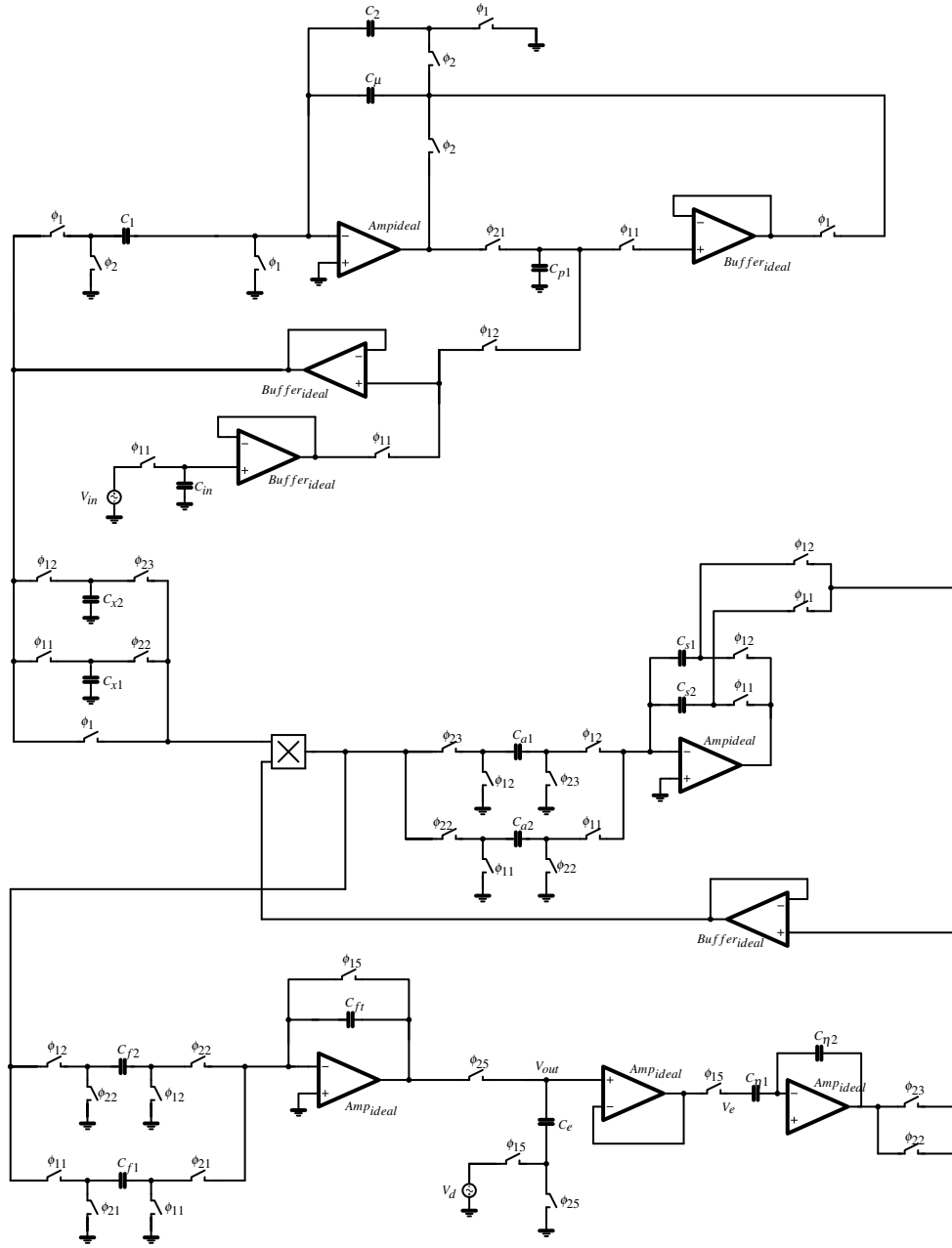


Figura 4.15: Circuito final ideal

4.4. O valor resultante da soma fica armazenado na capacidade C_e durante ϕ_{25} .

$$y(n) = \sum_{k=0}^K w_k x_k(n) \quad (4.4)$$

No instante ϕ_{15} , o valor da soma pesada, é comparado com o sinal desejado, V_d . A comparação é realizada no somador/subtractor da secção 4.5.3, onde o resultado é o erro de adaptação $e(n)$. Neste mesmo instante, este erro é multiplicado por uma constante η , que neste simulação tem o

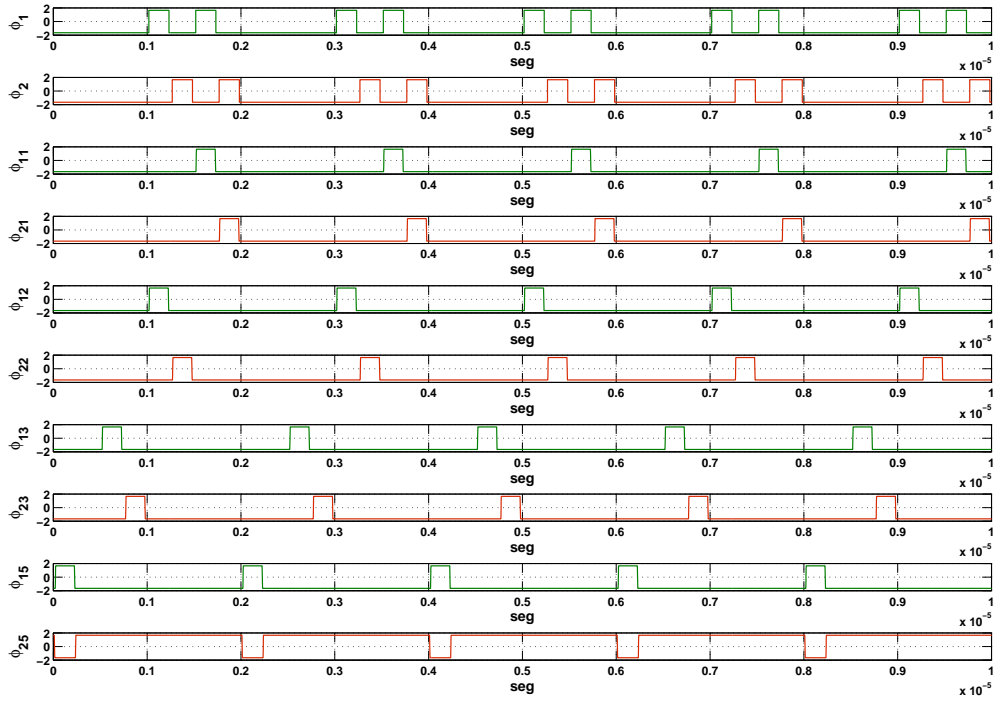


Figura 4.16: Sinais de controlo para circuito final ideal

Sinal	Período	Atraso	Tempo de Impulso	Valor do Impulso
ϕ_{11}	2000ns	2020ns	200n	1.65
ϕ_{21}	2000ns	2270ns	200n	1.65
ϕ_{11}	2000ns	1520ns	200n	1.65
ϕ_{21}	2000ns	1770ns	200n	1.65
ϕ_{12}	2000ns	1020ns	200n	1.65
ϕ_{22}	2000ns	1270ns	200n	1.65
ϕ_{13}	2000ns	520ns	200n	1.65
ϕ_{23}	2000ns	770ns	200n	1.65
ϕ_{15}	2000ns	20ns	200n	1.65
ϕ_{25}	2000ns	20ns	200n	-1.65

Tabela 4.2: Sinais de controlo

valor 0.2 resultante de $\frac{C_{\eta 1}}{C_{\eta 2}}$, onde $C_{\eta 1}$ é 2pF e $C_{\eta 2}$ é igual a 10pF. O valor da constante η permite definir a velocidade de convergência do sistema.

Como a equação para actualização dos coeficientes é dada por:

$$w_k(n) = w_k(n-1) + \eta e(n)x_k(n) \quad (4.5)$$

e já com valor do erro multiplicado pela constante η , é possível logo de seguida proceder aos

cálculos dos coeficientes do filtro. Estes cálculos são realizados nos instantes ϕ_{23} e ϕ_{22} , onde é multiplicado ηe por $x_k(n)$ que anteriormente foi memorizado nas capacidades C_{x1} e C_{x2} . O valor de cada multiplicação é amostrado nos mesmos instantes nas capacidades C_{a1} e C_{a2} , sendo posteriormente somado ao valor que estava armazenado nas capacidades C_{s1} e C_{s2} e enviado para o novo cálculo da soma pesada da saída do filtro nos instantes ϕ_{12} e ϕ_{11} , respectivamente.

Na figura 4.17 são ilustrados os sinais mais relevantes obtidos na adaptação. Na entrada do filtro foi aplicada uma onda sinusoidal, V_{in} , com amplitude $800mV$ e com frequência $10kHz$. O comportamento do sinal de saída do filtro adaptativo, V_{out} , foi tender para o valor da entrada, o que se traduz na minimização do erro representado por V_e . Na figura 4.18 é possível visualizar a variação dos coeficientes.

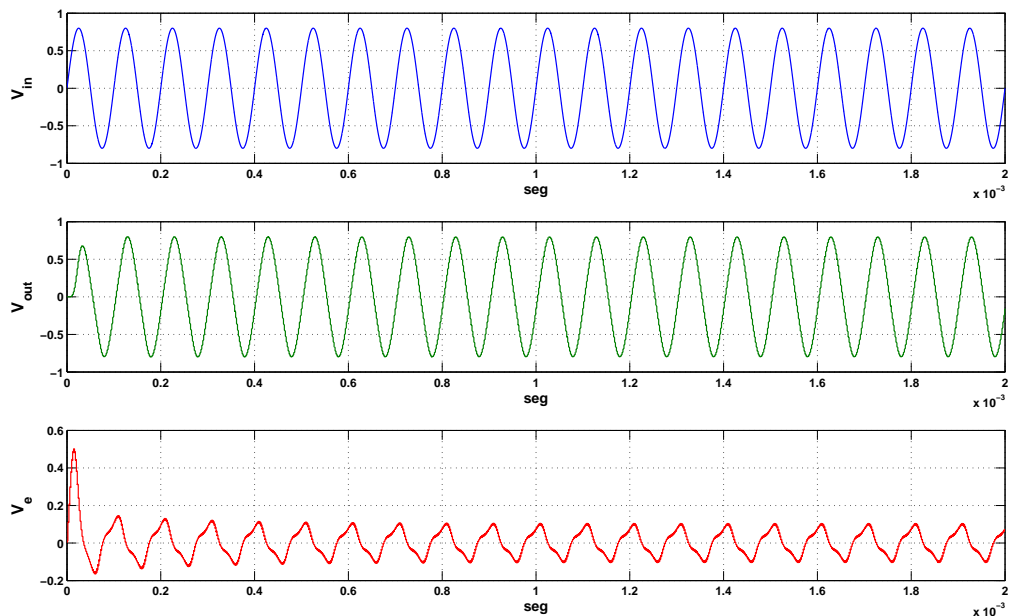


Figura 4.17: Entrada, saída e erro do filtro adaptativo ideal

As oscilações presentes, devem-se às capacidades parasitas existentes em cada nó no circuito.

4.7 Simulação em Matlab/Simulink do circuito final ideal

Os parâmetros atribuídos nesta secção são os mesmos da simulação da secção anterior. Na figura 4.19 os resultados obtidos são muito idênticos aos obtidos anteriormente, com excepção do erro, que neste caso tende mesmo para $0V$.

Comparando as convergências dos coeficientes desta simulação com a anterior, é possível visualizar que os valores finais seguem o mesmo comportamento. Os coeficientes da secção anterior, tentam aproximar-se dos obtidos na figura 4.20, não conseguindo no entanto atingir os mesmos

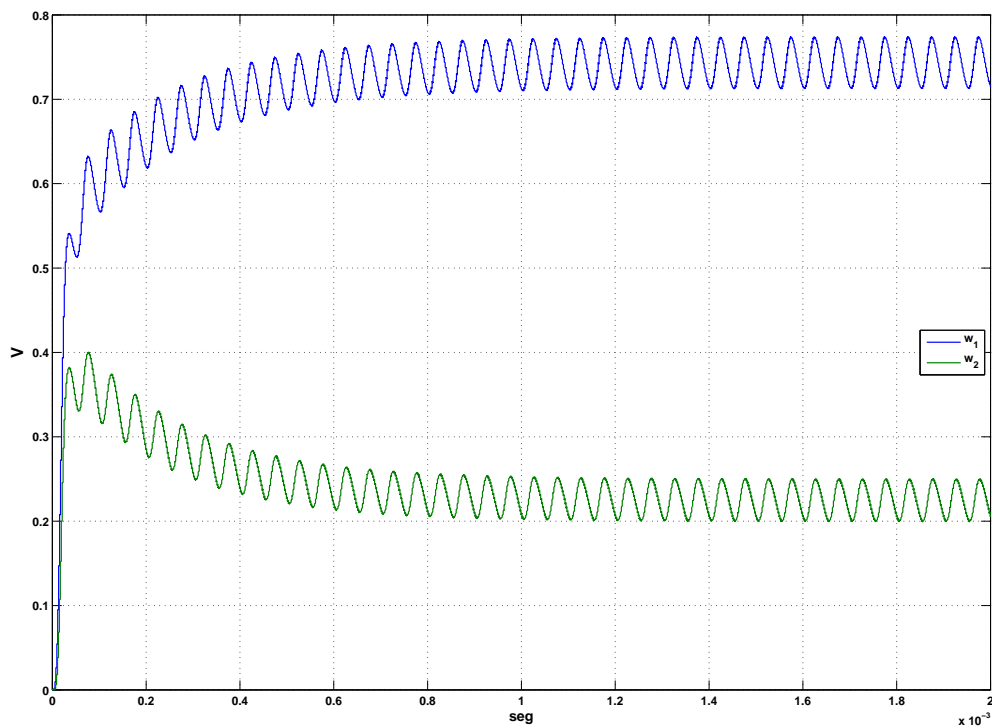


Figura 4.18: Coeficientes do filtro adaptativo ideal

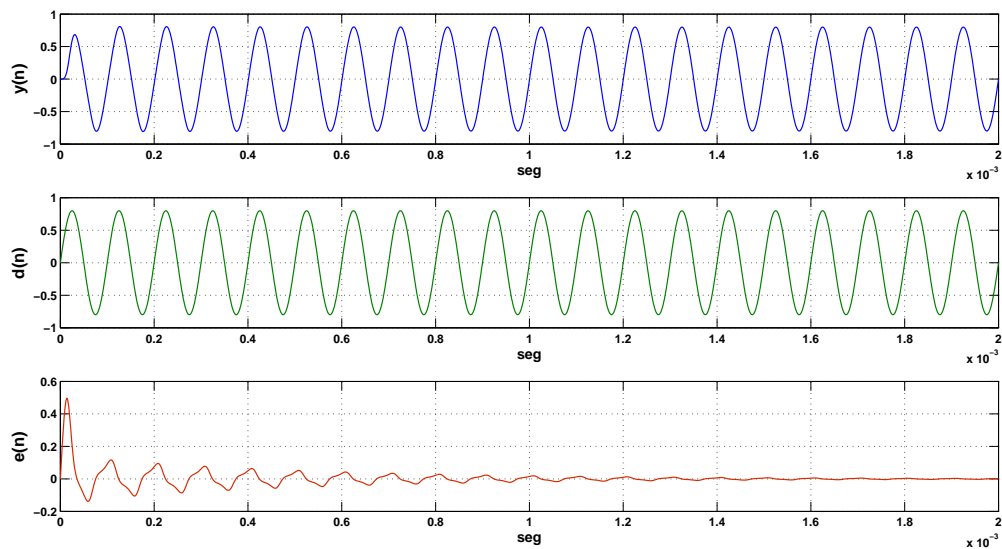


Figura 4.19: Sinais do filtro final em Matlab/Simulink

valores. É de notar que na simulação do circuito da secção anterior foram adicionadas capacidades parasitas em todos os nós, sendo estas responsáveis por erros nos valores finais devido a partilhas

de carga, o que justifica as diferenças encontradas.

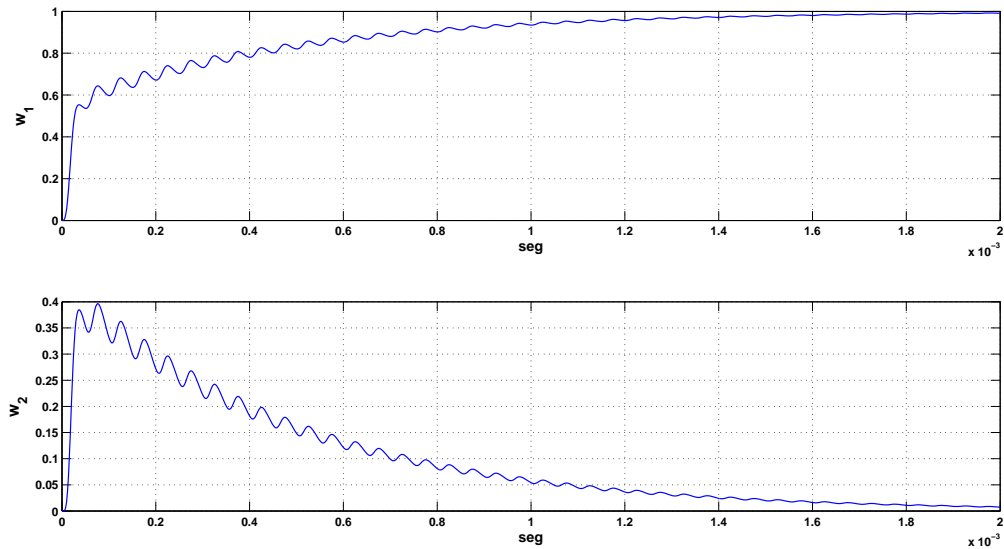


Figura 4.20: Coeficientes do filtro final em Matlab/Simulink

4.8 Implementação do Amplificador

Todas as simulações apresentadas aqui foram realizadas usando amplificadores ideais (embora de ganho finito - $10000V/V$). Nesta secção é ilustrada a implementação de um amplificador em CMOS-VLSI. O esquema do amplificador está ilustrado na figura 4.21 e representa um amplificador diferencial de dois andares para garantir um ganho em malha aberta de valor suficientemente elevado. A capacidade de $350fF$ é colocada para realizar uma compensação de Miller.

Na figura 4.22 é visível o ganho em malha aberta do amplificador, bem como a variação de fase. Para os resultados obtidos, os valores dos transístores podem ser consultados na tabela 4.3.

O amplificador simulado possui uma ganho em malha aberta de $76dB$, um produto ganho-largura de banda de aproximadamente $62MHz(V/V)$ e uma margem de fase de 60° .

Nome	Tipo	W(μm)	L(μm)
M_1	PMOS	5.8	1
M_2	PMOS	5.8	1
M_3	NMOS	12.78	1
M_4	NMOS	12.78	1
M_5	PMOS	100	1
M_6	PMOS	100	1
M_7	PMOS	300	1
M_8	NMOS	81.34	1

Tabela 4.3: Tamanhos dos transístores do amplificador

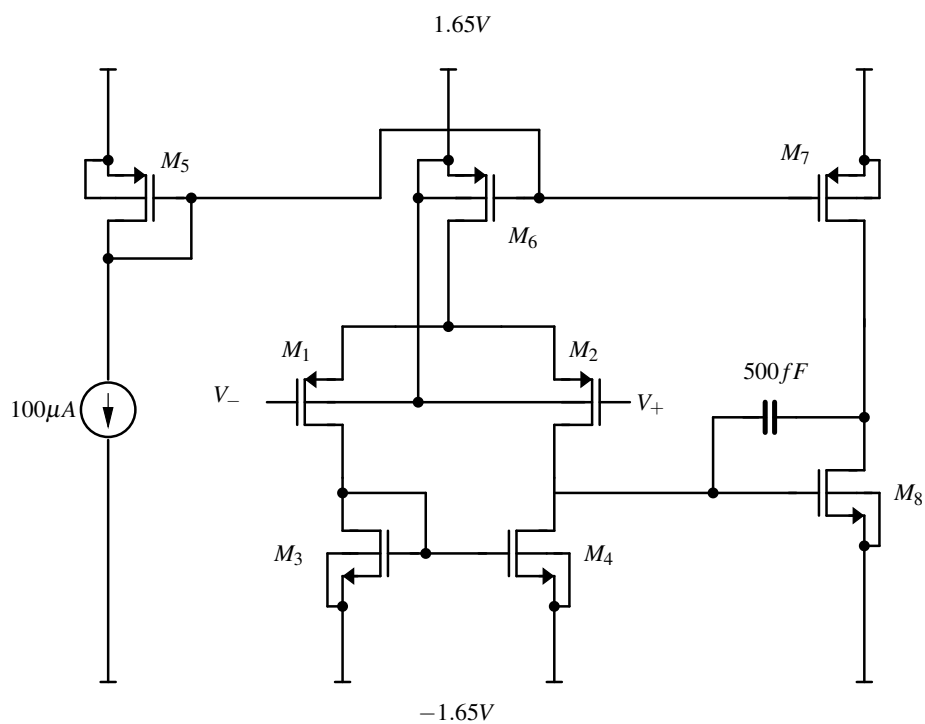


Figura 4.21: Circuito do amplificador

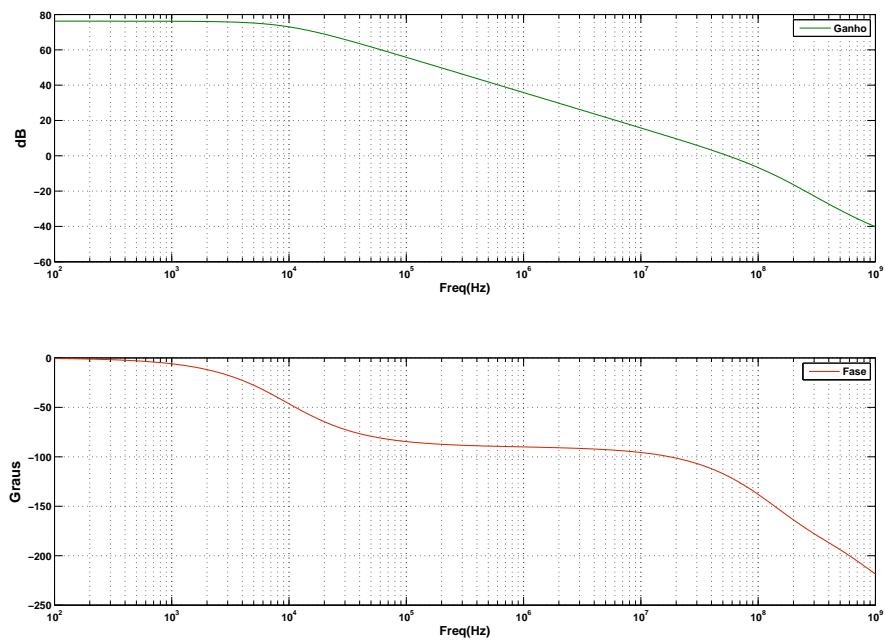


Figura 4.22: Ganho e fase do amplificador

4.9 Implementação do Switch

Os *switches* até aqui utilizados foram ideais, definidos para fecharem acima de $100mV$, colocando uma resistência de 0.1Ω , e para abrirem abaixo de $-100mV$, com uma resistência de $1T\Omega$. Para a implementação do *switch* foi usado o circuito da figura 4.23. A passagem do sinal é feita através transístor *NMOS* em paralelo com um *PMOS* para se garantir a máxima excursão dos sinais.

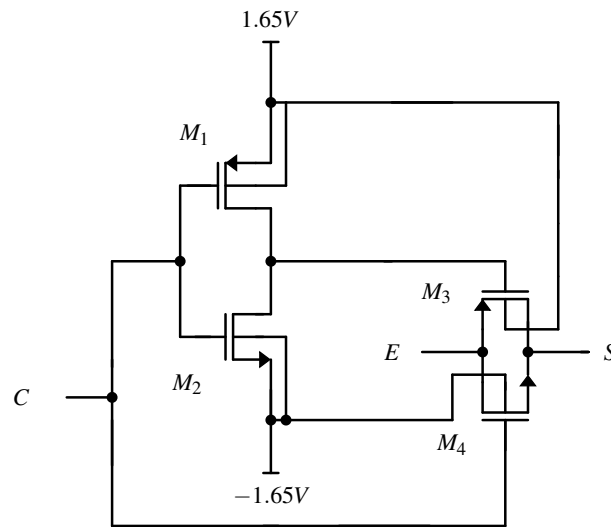


Figura 4.23: Circuito do *Switch*

Nome	Tipo	W(μm)	L(μm)
M_1	<i>PMOS</i>	5.8	1
M_2	<i>PMOS</i>	5.8	1
M_3	<i>NMOS</i>	12.78	1
M_4	<i>NMOS</i>	12.78	1

Tabela 4.4: Tamanhos dos transístores do switch

4.10 Simulações finais

Após o desenho de todos os blocos apresentados anteriormente, é necessário a interligação destes para comprovar o funcionamento do filtro adaptativo. Foram realizadas duas simulações, uma com dois coeficientes e outra com quatro coeficientes, com as fontes, sinais de controlo e outros parâmetros, iguais aos atribuídos na secção 4.6.

4.10.1 Simulação com dois coeficientes

Os resultados da figura 4.24 comprovam o funcionamento do filtro adaptativo com a interligação de todos os blocos implementados.

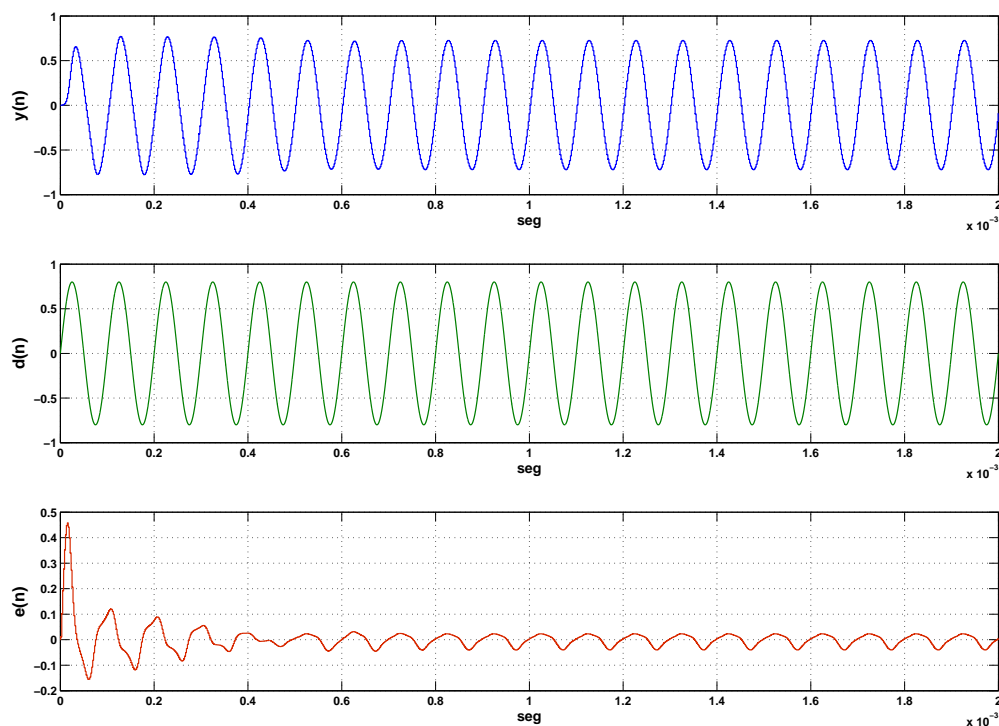


Figura 4.24: Sinais do circuito final, dois coeficientes

Nesta figura a saída do filtro conseguiu igualizar-se com sinal de entrada. Mais uma vez, o sinal erro não consegue tender para o valor 0V, devido às capacidades parasitas, mas o sinal de saída aprendeu rapidamente o sinal de entrada. A variação dos coeficientes está presente na figura 4.25. Observando a variação dos coeficientes, repara-se que um deles satura. Isto deve-se à existência de *offset's* no circuito provenientes dos amplificadores e eventualmente de CFT (*clock feedthrough*).

4.10.2 Simulação com quatro coeficientes

Simulando o filtro adaptativo, agora com quatro coeficientes observa-se que o sinal de saída consegue aprender o sinal desejado. O sinal de saída converge rapidamente para o valor introduzido na entrada, como atesta a figura 4.26.

Mais um vez é visível na figura 4.27 o problema enunciado na simulação anterior referente aos *offset's* no circuito.

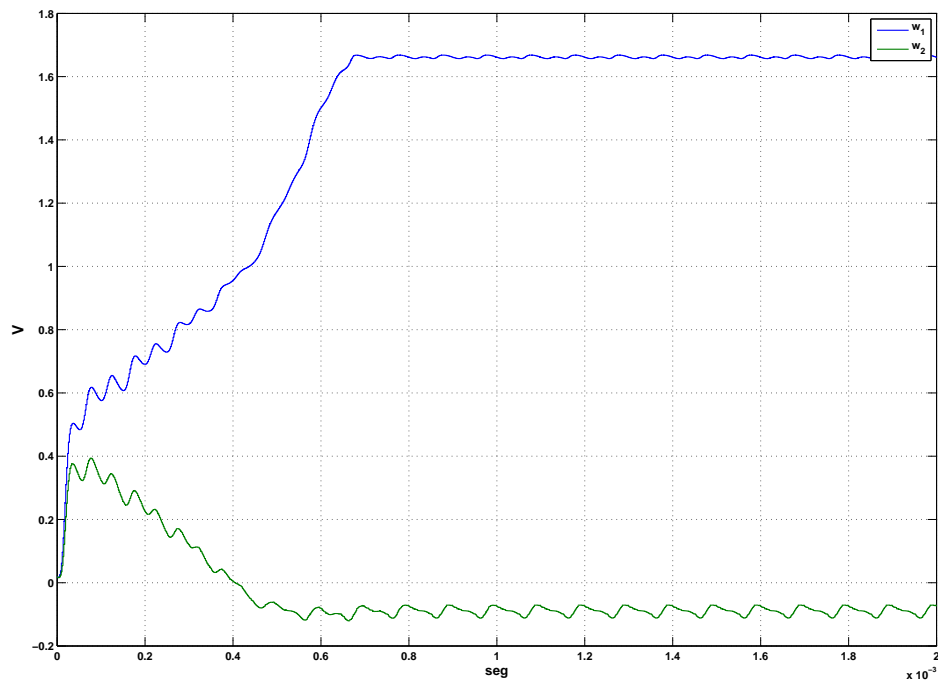


Figura 4.25: Coeficientes do circuito final, dois coeficientes

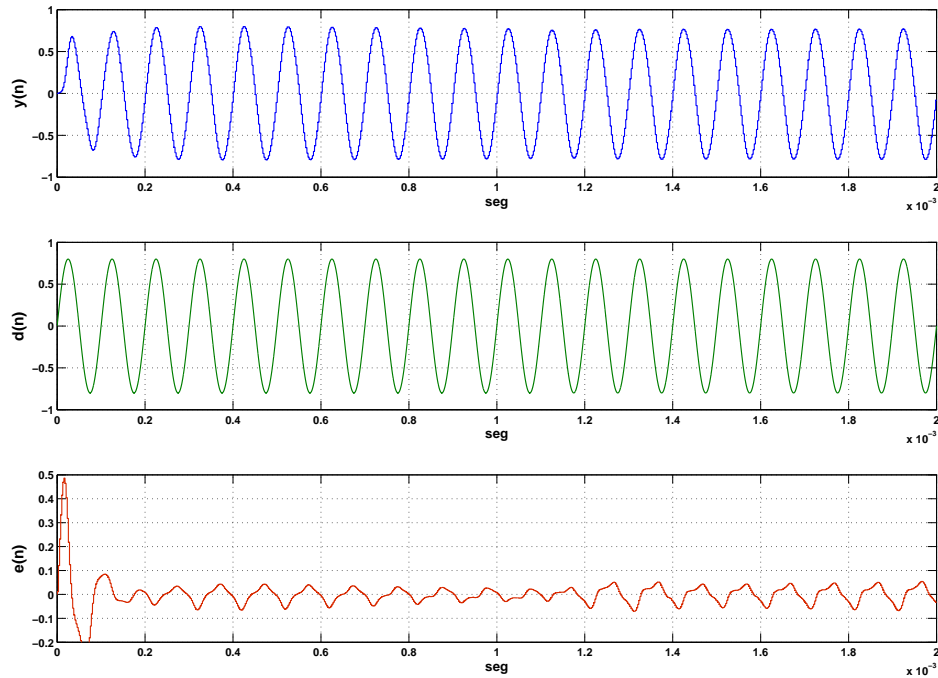


Figura 4.26: Sinais do circuito final, quatro coeficientes

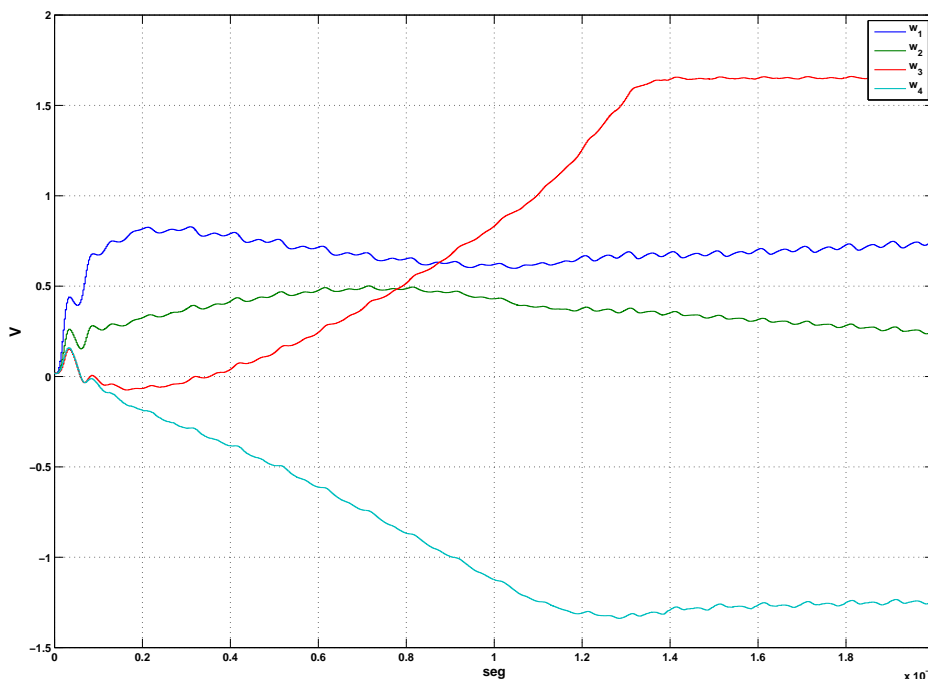


Figura 4.27: Coeficientes do circuito final, quatro coeficientes

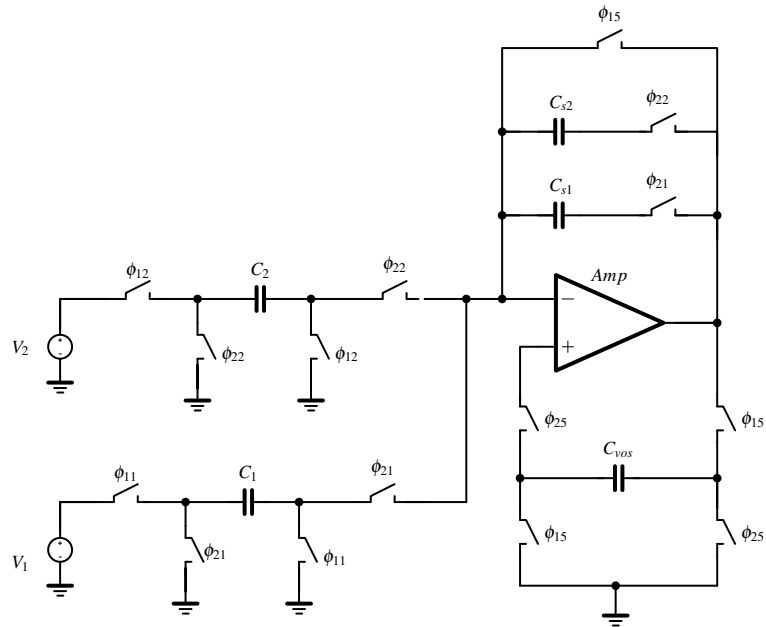
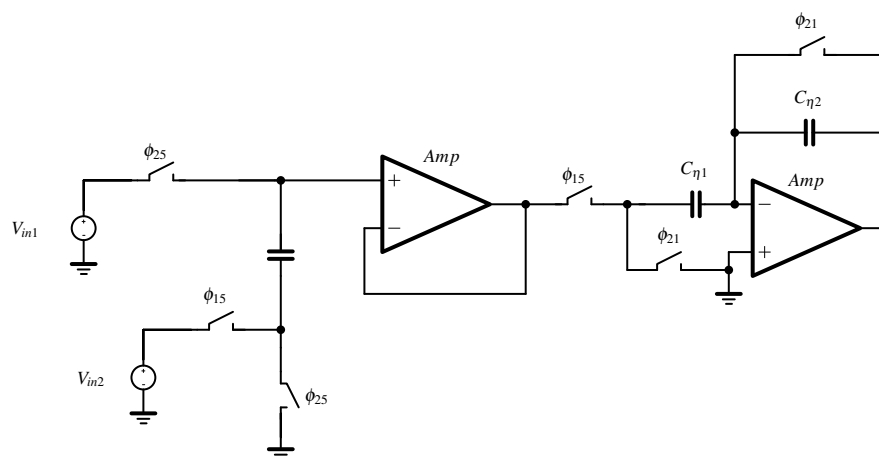
4.11 Compensação dos *offsets*

A forma pensada para minimizar os *offsets* no sistema acrescentou-se ao circuito integrador (4.5.2) e ao circuito subtrator (4.5.3) uma capacidade. Esta capacidade, juntamente com os *switches*, vai armazenar num instante o valor do *offset* existente no amplificador, de forma a num instante posterior o valor guardado vá subtrair ao valor do *offset*.

A compensação de *offset* realizado no integrador esta ilustrada na figura 4.28 e do subtrator na figura 4.29.

Analisando os resultados da simulação nas figuras 4.30 e 4.31, agora com compensação de *offset*, reparamos que mesmo assim um dos coeficientes satura. Isto acontece porque o valor que é armazenado na capacidade não é exactamente igual ao valor de *offset*. Durante as comutações dos *switches*, existe uma ligeira perda de carga no condensador. Por muito pequeno que seja esta diferença, ao longo do tempo, este valor vai sendo incrementado.

Este problema é conhecido e uma forma de o contrariar pela utilização do algoritmo *Leaky LMS* [18] [19]. Este algoritmo consegue ter uma relação de compromisso entre minimizar o erro quadrático médio e minimizar o valor dos coeficientes, mantendo-os sempre numa região linear. A introdução deste algoritmo implica um menor desempenho do sistema, adicionando pouca complexidade ao hardware, mas com a garantia de funcionalidade. Uma forma para tentar minimizar a complexidade poderia passar pelo uso do algoritmo *Sign LMS* [20].

Figura 4.28: Compensação de *offset* no integradorFigura 4.29: Compensação de *offset* no subtrator

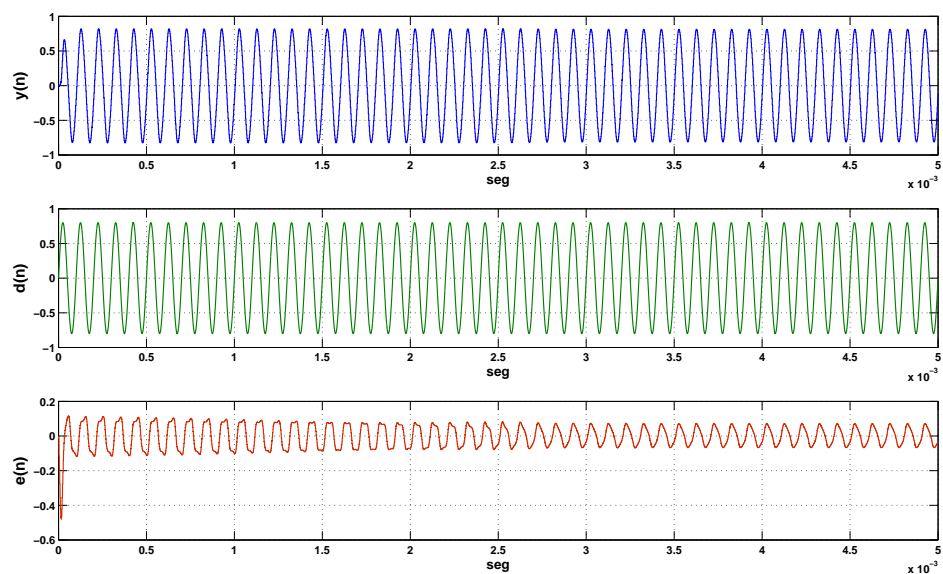


Figura 4.30: Coeficientes do circuito final, quatro coeficientes

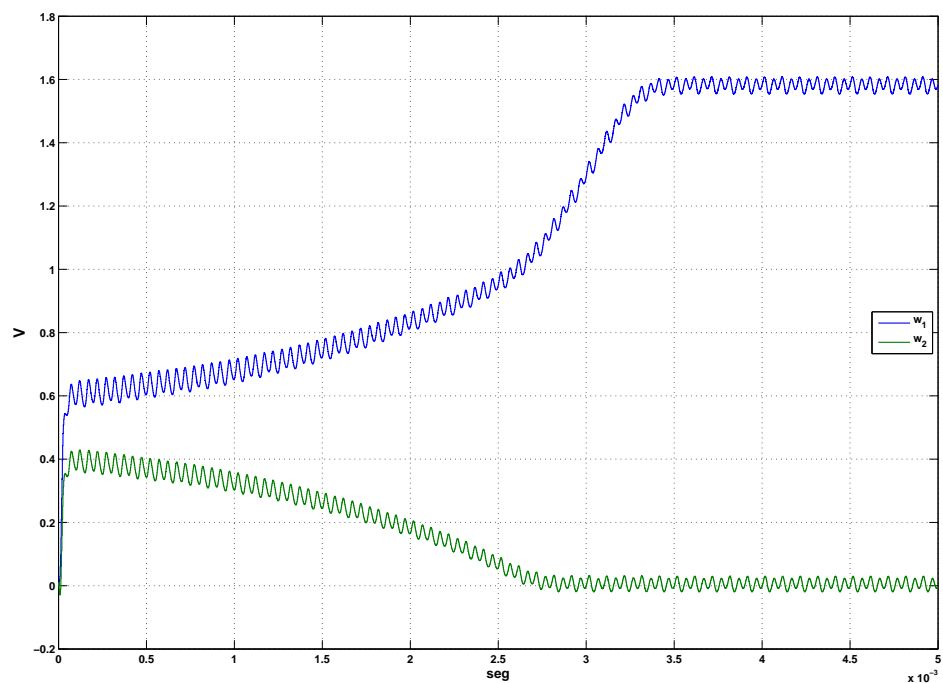


Figura 4.31: Coeficientes do circuito final, quatro coeficientes

Capítulo 5

Conclusões e Trabalho Futuro

5.1 Conclusão

Neste trabalho foi implementado um filtro analógico adaptativo, utilizando condensadores comutados. O circuito implementado é constituído por um filtro Gamma que juntamente com o algoritmo LMS, implementam um filtro adaptativo.

Conseguiu-se com sucesso multiplexar o filtro Gamma. A multiplexagem inicia-se a partir da última tap do filtro, até à entrada do sinal num processo iterativo.

O algoritmo LMS foi o utilizado porque a sua implementação em CMOS torna-se mais fácil. Para comprovar o funcionamento do filtro adaptativo foram implementadas várias simulações no *software Matlab/Simulink*, que durante o trabalho foram comparadas com o circuito implementado em *Cadence* utilizando o filtro Gamma multiplexado. Iniciou-se a implementação recorrendo a componentes ideais, sendo mais tarde substituídos por circuitos implementados. Foram propostos três somadores, um para realizar o somatório pesado dos vários atrasos do filtro Gamma, um somador integrador para calcular a variação dos coeficientes do filtro e um somador para calcular a diferença entre o sinal desejado e a saída do filtro. Além dos somadores, foi implementado um amplificador para ser utilizado no filtro, nos somadores e para funcionar como *buffer*.

Após a interligação dos blocos implementados, foram realizadas simulações finais onde foi encontrado o problema dos *offsets* durante a adaptação. Este problema provoca a saturação dos coeficientes, apesar de visualizarmos que a saída do filtro consegue aprender o sinal desejado. Este problema é resolvido com ligeira alteração no algoritmo LMS, passando-o para *Leaky LMS*, e com circuitos de minimização de *offset*. Como este novo algoritmo é mais complexo, pode-se juntar o *Sign LMS* de forma a reduzir a complexidade.

5.2 Trabalho Futuro

Um dos blocos importantes neste circuito é um multiplicador que até à escrita desta dissertação não foi implementado. Este é um dos blocos deverá ser estudado num futuro próximo, devendo ainda proceder-se ao cancelamento do *offset* através da alteração de circuitos juntamente com o *Leaky LMS*. Poderá ainda proceder-se ao desenho de um novo amplificador utilizando um *folded-cascode* para melhorar a largura de banda.

No artigo [17] é exposta a forma de como o μ do filtro Gamma pode variar adaptativamente. A introdução deste novo coeficiente, melhora em muito a aprendizagem pois a resolução da filtragem vai variar consoante o sinal que é filtrado. Desta forma dependendo do sinal de entrada, o filtro consegue encontrar a melhor profundidade de memória para o sistema adaptativo. No anexo A é exposta uma breve descrição de como juntar este bloco ao circuito.

Para finalizar deverá proceder-se ao desenho *layout* do circuito final.

Anexo A

Implementação futura

Neste anexo é ilustrado matematicamente a forma como variar a constante μ já existente no filtro Gamma. Este coeficiente vai permitir ao filtro Gamma conseguir variar a sua resolução/profundidade consoante o sinal de entrada.

A.1 Variação do μ

Na secção 2.5 deste documento é exposta a forma de calcular a variação do coeficientes \vec{w}_k . Da mesma forma, através do cálculo do erro total do sinal do sistema:

$$E = \sum_{t=0}^T E_t = \sum_{t=0}^T \frac{1}{2} e^2(n) = \sum_{t=0}^T \frac{1}{2} [d(n) - y(n)]^2 \quad (\text{A.1})$$

Reparamos que a correcção dos coeficientes varia proporcionalmente para o valor negativo do gradiente associado:

$$\Delta w_k = -\eta \frac{\partial E}{\partial w_k} \quad (\text{A.2})$$

ou

$$\Delta \mu = -\eta \frac{\partial E}{\partial \mu} \quad (\text{A.3})$$

onde η é um factor de atenuação.

Desenvolvendo neste caso $\Delta \mu$,

$$\Delta \mu = -\eta \frac{\partial E}{\partial \mu} = \eta \sum_{t=0}^T e_t \sum_{k=0}^K w_k \frac{\partial x_k(n)}{\partial \mu} = \eta \sum_{t=0}^T e_t \sum_{k=0}^K w_k \alpha_k(n) \quad (\text{A.4})$$

onde $\alpha_k(n) = (\frac{\partial x_k(n)}{\partial \mu})$ e $\alpha_0(n) = 0$ Desta forma $\alpha_k(n)$ é dado por:

$$\alpha_k = (1 - \mu) \alpha_k(t-1) + \mu \alpha_{k-1}(t-1) + [x_{k-1}(t-1) - x_k(t-1)], k = 1, \dots, K \quad (\text{A.5})$$

As actualizações dos coeficientes μ podem ser aproximado a:

$$\Delta\mu = \eta \sum_{k=0}^K e(n)w_k\alpha_k(n) \quad (\text{A.6})$$

Referências

- [1] C.C. Took and D.P. Mandic. The quaternion lms algorithm for adaptive filtering of hyper-complex processes. *Signal Processing, IEEE Transactions on*, 57(4):1316–1327, April 2009.
- [2] Murat Çabuk. Adaptive step size and exponentially weighted affine projection algorithms. Tese de mestrado, Boğaziçi University, 2002.
- [3] National Instruments. Adaptive filter concepts (adaptive filter toolkit), Junho 2008. http://zone.ni.com/reference/en-XX/help/372357A-01/lvaftconcepts/aft_concepts_intro/.
- [4] H. Yoo, D.V. Anderson, and P. Hasler. On delay structures for the analog adaptive filters with long filter taps. *Signals, Systems and Computers, 2003. Conference Record of the Thirty-Seventh Asilomar Conference on*, 2:2021–2025 Vol.2, Nov. 2003.
- [5] H.P. Meana, L.N. de Rivera, and M.N. Miyatake. Imperfect integration and dc offset effects in the convergence performance of gradient search based analog adaptive filters. *Circuits and Systems, 1999. 42nd Midwest Symposium on*, 2:622–625 vol. 2, 1999.
- [6] T. Ndjountche, R. Unbehauen, and Fa-Long Luo. Dynamic analog resonator-based adaptive filters. *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, 3:161–164 vol.3, 2000.
- [7] Yan Xie and B.M. Al-Hashimi. Analogue adaptive filters using wave synthesis technique. *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, 1:I-849–52 Vol.1, May 2004.
- [8] M. Jafaripناه, B.M. Al-Hashimi, and N.M. White. Adaptive sensor response correction using analog filter compatible with digital technology [load cell sensor applications]. *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pages 5389–5392 Vol. 6, May 2005.
- [9] E. Bertran. A fully analog adaptive-disturbance canceller. *Instrumentation and Measurement, IEEE Transactions on*, 56(5):1605–1609, Oct. 2007.
- [10] Jui-Kuo Juan, J.G. Harris, and J.C. Principe. Analog hardware implementation of adaptive filter structures. *Neural Networks, 1997., International Conference on*, 2:916–921 vol.2, Jun 1997.
- [11] E.J. McDonald and B.A. Minch. Synthesis of a translinear analog adaptive filter. *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, 3:III-321–III-324 vol.3, 2002.

- [12] E.J. McDonald and B.A. Minch. Multi-level simulation of a translinear analog adaptive filter. *Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02). IEEE International Conference on*, 4:IV-3992-IV-3995 vol.4, 2002.
- [13] Sundar G Sankaran. On ways to improve adaptive filter performance. Tese de douturamento, Virginia Tech., 1999.
- [14] Paulo Alexandre C. Marques. *Introdução à Filtragem Adaptativa*. ISEL – Instituto Superior de Engenharia de Lisboa, 2003.
- [15] Bert De Vries. Temporal processing with neural networks – the development of the gamma model. Tese de douturamento, University of Florida, 1991.
- [16] Jyh-Ming Kuo and S. Celebi. Adaptation of memory depth in the gamma filter. *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, iv:IV/373-IV/376 vol.4, Apr 1994.
- [17] J.C. Principe, B. de Vries, and P.G. de Oliveira. The gamma-filter-a new class of adaptive iir filters with restricted feedback. *Signal Processing, IEEE Transactions on*, 41(2):649-656, Feb 1993.
- [18] C.-P.J. Tzeng. An adaptive offset cancellation technique for adaptive filters. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 38(5):799-803, May 1990.
- [19] J.G. Harris, Jui-Kuo Juan, and J.C. Principe. Analog hardware implementation of continuous-time adaptive filter structures. *Analog Integrated Circuits and Signal Processing, IEEE Transactions on*, 18(2):209-227, Feb 1999.
- [20] E. Cetin, I. Kale, and R.C.S. Morling. On various low-hardware-complexity lms algorithms for adaptive i/q correction in quadrature receivers. In *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, volume 4, pages IV-461-4 Vol.4, May 2004.
- [21] Bernard Widrow and Eugene Walach. *Adaptive inverse control*. Cambridge University Press, 1996.
- [22] Sílvio A. Abrantes. *Processamento adaptativo de sinais*. Fundação Calouste Gulbenkian, 2000.
- [23] Simon Haykin. *Adaptive filter theory*. Prentice Hall International, 2nd ed edition, cop. 1996.
- [24] Odile Macchi. *Adaptive Processing The Least Mean Squares Approach with Applications in Transmission*. John Wiley & Sons, cop. 1995.